APPLICATION NOTE

# SH-4 Interface to SDRAM

## Introduction

This application note has been written to aid designers connecting Synchronous Dynamic Random Access Memory (SDRAM) to the Bus State Controller (BSC) of SH7750S and SH7751. The application note starts with the principles of SDRAMs. This gives us the foundation to explain the details of the slight different BSC of these products. After a presentation of the physical connections and the timings an example closes the discussion.

For designers who wish to set up the BSC as quickly as possible, it is recommended that the section 'Worked Example – EDS2516ACTA-7A connected to 2x256-Mbit (64 Mbytes) SDRAM @ 100 MHz bus frequency' is used as the starting point. More detailed information can be gathered from the other sections when necessary.

## Contents

1 SDRAM INTRODUCTION ..... 5
1.1 DRAM HISTORY ..... 5
1.2 DRAM CELL ..... 5
1.2.1 THE BASIC CELL ..... 5
1.2.2 SENSE AMPLIFIER ..... 6
1.2.3 BIT RESTORING ..... 6
1.2.4 WRITE AND REFRESH ..... 6
1.2.5 PRECHARGE ..... 7
1.3 CLOCKED INTERFACE ..... 7
1.4 SPLIT-BANK ARCHITECTURE ..... 9
1.5 PROGRAMMABILITY ..... 9
1.6 INDIVIDUAL BYTE ENABLES ..... 11
1.7 TWO REFRESH MODES ..... 12
1.8 MARKING OF SDRAMS ..... 12
2 THEORY OF SDRAM OPERATION ..... 13

2.1 OVERVIEW ..... 13
2.2 POWER-ON SEQUENCE ..... 14
2.3 COMMAND OPERATION ..... 15
2.4 PRECHARGE OPERATIONS ..... 20
2.5 MODE REGISTER ..... 21
2.6 ACTV COMMAND SEQUENCE ..... 25
2.7 READ OPERATION ..... 25
2.8 WRITE OPERATION ..... 28
2.9 STATE MACHINE ..... 31
3 SH-4 BUS STATE CONTROLLER SDRAM OVERVIEW ..... 34
4 SDRAM CONTROLLER DIFFERENCES BETWEEN SH7750S & -7751 ..... 37
4.1 OVERVIEW ..... 37
4.2 BUS STATE CONTROLLER REGISTER ..... 39
4.2.1 BUS CONTROL REGISTER 1 (BCR1): ..... 39
4.2.2 BUS CONTROL REGISTER 2 (BCR2): ..... 39
4.2.3 WAIT CONTROL REGISTER 1 (WCR1): ..... 39
4.2.4 WAIT CONTROL REGISTER 2 (WCR2): ..... 39
4.2.5 WAIT CONTROL REGISTER 3 (WCR3): ..... 40
4.2.6 INDIVIDUAL MEMORY CONTROL REGISTER (MCR): ..... 40
4.2.7 PCMCIA CONTROL REGISTER (PCR): ..... 41
4.2.8 REFRESH TIMER CONTROL/STATUS REGISTER (RTSCR): ..... 41
4.2.9 REFRESH COUNT REGISTER (RFCR): ..... 42
4.2.10 REFRESH TIMER COUNTER REGISTER (RTCNT): ..... 42
4.2.11 REFRESH TIME CONSTANT REGISTER (RTCOR): ..... 42
4.2.12 SYNCHRONOUS DRAM MODE REGISTER 2 & 3(SDMR2, -3): ..... 42
5 CONNECTING THE SH-4 DIRECTLY TO SDRAM ..... 44
5.1 PHYSICAL CONNECTION ..... 44
5.1.1 ADDRESS MULTIPLEXING ..... 45

5.2 TIMING CHECK ..... 47
5.2.1 SETUP AND HOLD TIMES ..... 48
5.2.2 WAIT STATES ..... 48
5.2.3 AUTO REFRESH INTERVAL TIMING ..... 48
5.2.4 CLOCK QUALITY ..... 48
5.2.5 SDRAM SETUP AND HOLD PARAMETERS ..... 50
5.2.5.1 TAC ..... 50
5.2.5.2 TOH ..... 50
5.2.5.3 TSI ..... 50
5.2.5.4 THI ..... 50
5.2.5.5 tLZ ..... 52
5.2.5.6 tHZ ..... 52
5.2.6 SDRAM WAIT STATE PARAMETERS ..... 54
5.2.6.1 TRRD ..... 54
5.2.6.2 TRCD ..... 54
5.2.6.3 TRP ..... 54
5.2.6.4 TRAS (1) ..... 54
5.2.6.5 TDPL ..... 55
5.2.6.6 TDAL ..... 55
5.2.6.7 TRC ..... 55
5.2.6.8 TROH ..... 56
5.2.6.9 TCCD ..... 56
5.2.6.10 TDWD/TDQM ..... 56
5.2.6.11 TDQZ ..... 56
5.2.6.12 TSRX/$I_{SREX}$ ..... 57
5.2.6.13 TCKE ..... 57
5.2.6.14 TMRD ..... 57
5.2.6.15 $I_{EP}$ 57
5.2.6.16 $I_{APR}$ 57
5.2.6.17 $I_{CDD}$ 58
5.2.6.18 $I_{PEC}$, $I_{BSR}$, $I_{BSH}$ & $I_{BSW}$ ..... 58
5.2.7 AUTO REFRESH INTERVAL TIMING ..... 58
5.2.7.1 $T_{REF}$ ..... 58
5.2.7.2 TRAS (2) ..... 58
5.2.8 CLOCK QUALITY PARAMETERS ..... 59
5.2.8.1 tT ..... 59
5.2.8.2 TCLK ..... 61
5.2.8.3 TCH/TCL ..... 61
5.2.9 TIMING SHEET, SH7750S AND 256-MBIT SDRAM ..... 62
5.3 OTHER LOADING ISSUES ..... 64

5.3.1 GENERAL BUS LOADING – OUTPUT BUFFER TIMING ..... 64
5.3.2 GENERAL BUS LOADING – INPUT SIGNAL TIMING ..... 65
5.3.3 CLOCK CONNECTION GUIDELINES ..... 65
5.3.3.1 CLOCK TOPOLOGY ..... 65
5.3.3.2 CKIO LOAD ..... 66
5.3.3.3 CHARACTERISTIC IMPEDANCE ..... 67
5.3.3.4 TERMINATION ..... 67
5.3.3.5 TRACK LENGTH ..... 68
6 WORKED EXAMPLE – EDS2516ACTA-7A CONNECTED TO 2X256-MBIT (64 MBYTES) SDRAM @ 100 MHZ BUS FREQUENCY ..... 69
6.1 PHYSICAL CONNECTION ..... 69
6.2 TIMING CHECK ..... 72
6.2.1 SETUP AND HOLD PARAMETERS ..... 72
6.2.2 WAIT STATE PARAMETERS ..... 74
6.2.3 AUTO REFRESH INTERVAL PARAMETERS ..... 75
6.2.4 CLOCK QUALITY PARAMETERS ..... 76
6.3 PRODUCING THE INITIALISATION CODE ..... 77
6.3.1 EXAMPLE SDRAM INITIALISATION CODE ..... 78
7 SUMMARY ..... 79
8 REFERENCES ..... 79

# 1 SDRAM Introduction

## 1.1 DRAM History

For many years the Dynamic Random Access Memory (DRAM) has been the standard by which computer memory systems have been designed. During this time DRAM architectures have found their way into virtually every market segment, from low-end personal computers to supercomputer systems costing millions of dollars.

The main advantage of a DRAM is the package densities, which can be attained due to the fact that single memory cells are represented by a single transistor and capacitor as opposed to a series of transistors. These densities allow for large amounts of memory to be represented by relatively few components. Advances in semiconductor process technology have caused package densities to increase by a factor of 1000 since the modern DRAM architecture was first introduced in the late 1970s, from 16-kbit devices then to 512-Mbit devices today.

Although package and process technologies have improved, the basic architecture of the DRAM has not changed since it was first introduced. A $2^n$-bit DRAM is typically organised as $2^n/2$ rows by $2^n/2$ columns. The rows are addressable using the Row Address Select (RAS) signal, and the columns are addressable using the Column Address Select (CAS)signal. More importantly, the speed with which data can be retrieved from a DRAM has not kept pace with increases in microprocessor speeds. This is no easy task since in some cases microprocessor speeds have increased by an order of magnitude just since the late 1980s.

A finite period of time is required by the DRAM to access the requested location once the address and proper control signals have been asserted. Over the years various flavours of the basic DRAM architecture have been introduced in order to help reduce these access times.

However, the end result has been that the processor must spend increasing amounts of time waiting for data.

## 1.2 DRAM Cell

### 1.2.1 The Basic Cell

The basic DRAM cell is comprised of a transistor (as a switch) and a capacitor (as a data storage element). The digit that is saved in the storage cell, is determined as logic 0, or 1, by the voltage potential stored inside the capacitor C. If the voltage stored is Vcc, then it means a logic 1. If the potential is ground, then the logic 0 is inside.

Reading of the storage cell is activated by a WORD LINE and a pair of BIT LINES. The WORD LINE controls the switch of the transistor T to turn on, in order to let the charge on the capacitor pass to the bit-line BL. The charge on the capacitor C and the Bit line BL will share together to form a new voltage potential. Before the open of transistor T, the bit-lines BL and BL* need to be precharged to the same voltage level 1/2Vcc. After the transistor T is opened, bit-line BL will get a new voltage potential, due to the voltage sharing, no matter that the C is with Vcc or with Ground potential. Then the bit line pairs, that carry differential voltages, will be connected to the sense amplifier circuit to determine if 0 or 1 exists in that cell.

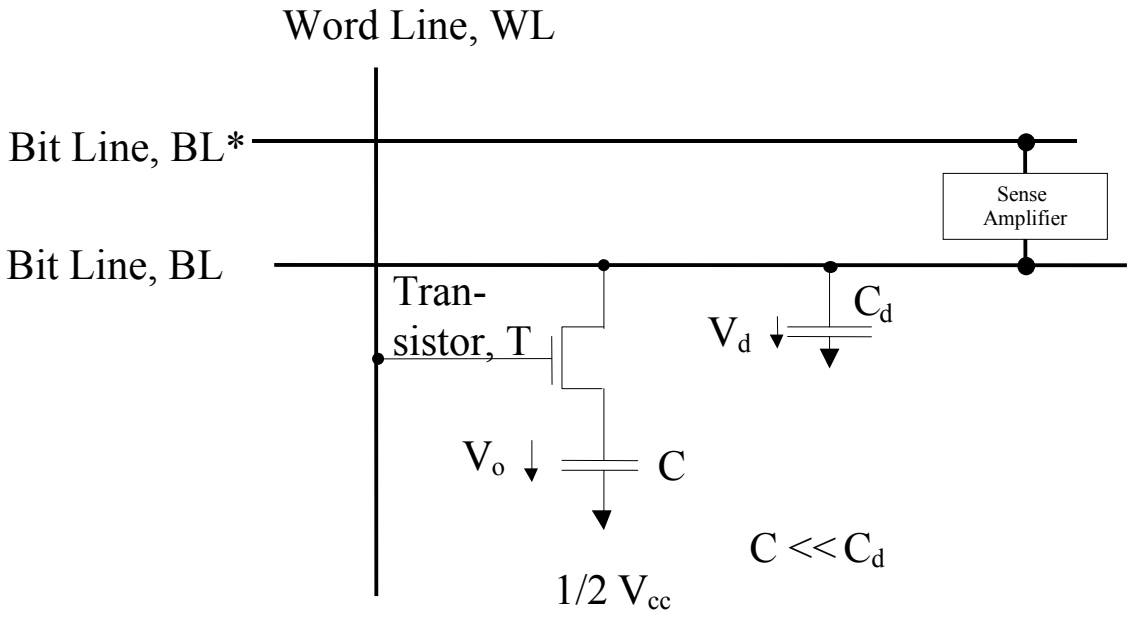


**Figure 1-1 DRAM Cell**

### 1.2.2 Sense Amplifier

After the charge on the BL bit-line becomes a newer potential value, the sense amplifier will detect the voltage difference between the bit-line pairs, and then force the voltages on the two line to reach the saturated Vcc and Ground potential. It's a kind of "splitting" between the original dual 1/2 Vcc voltage lines. At this moment, the content inside the DRAM storage cell is detected, and will be sent out to the output ports. However, the voltage potential in the cell capacitor is damaged, changing from Vcc or Gnd to 1/2 Vcc + (-)0.1V potential.

### 1.2.3 Bit Restoring

After the data of cell is sensed and latched by the sense amplifier, the BL bit-line will be latched and hold as the original cell binary state:

- if logical 1 (Vcc), bit-line BL will charge the cell capacitor and pull up its voltage to Vcc.
- if logical 0 (Gnd), the cell capacitor C will be discharged and its potential would be pulled down to ground potential.

This operation is quite helpful, while in the charge sharing of the cell capacitor, the voltage potential has been changed. The original digit of the cell capacitor becomes recovery before the end of the access cycle, because of its latched data line.

### 1.2.4 Write and Refresh

The previous discussion is only focused on the reading action of the DRAM bit cell. If we want to perform the writing on the bit cell, it is quite easy to accomplish that work. Just simply place the data on the bit lines BL, BL*, through the another two lines that are connected to the bit

lines. When the voltages on the bit lines approach the saturated Vcc and Gnd potential, the sense amplifier will automatically save the data into the storage cell, as the restoring process goes. Due to the fact that there is some leakage current existing on the capacitor of the DRAM cell, the bit data cannot be always kept in the cell for a long time without losing voltage potential. It is necessary that the content of the bit cell is kept being updated periodically. We call that work as refreshing. The way to refresh the bit cell is quite similar to read the data from the cell. Just open all the word lines and let the sense amplifier do the restoring work on every cell.

### 1.2.5 Precharge

It is very important that the bit lines of the DRAM are kept in the potential of 1/2Vcc. This is called as precharging of the DRAM. Precharging the bit lines to 1/2Vcc is the most fundamental step for all the commands or operations of the DRAM. Before reading or writing the bit cell, the bit lines need to be precharged.

## 1.3 Clocked Interface

Because of the basic architecture of the DRAM, once the processor has requested data, it must wait for a certain length of time until data returns. This can take some clock cycles, depending on the design of the memory system. The processor generates an address which is latched resulting in the assertion of RAS# and CAS#. RAS# must be held active throughout the entire access, meaning that the DRAM cannot accept new address and control information during this time.

This non-pipelined approach can have a very negative effect on the overall performance of a system. To solve this problem, memory systems have become increasingly complex, with a form of pipelining being implemented by multiple banks, which switch at different times in order to hide these inherent latencies. One bank of DRAMs is accessed, then another, and so on. A four-way interleaved memory system can offer some performance boost, but large amounts of real estate are normally required to facilitate this. In addition, DRAM control logic complexity is greatly increased.

Herein lies the biggest advantage of the SDRAM architecture. All control signals are sampled on the rising edge of the clock. Control signals such as RAS# and CAS# need only be asserted for one clock, meaning that the processor need only drive address and control signals for as long as it takes for this information to be recognised by the SDRAM controller. Unlike standard DRAMs, where CAS# could not be asserted on consecutive clocks, the SDRAM can accept a new column address on every clock, allowing for a memory system which can meet the throughput requirements of today's high-speed microprocessors.

The programmability offered by SDRAMs allows the processor to know how long it will take to retrieve the data once the memory cycle has been initiated. For example, assuming a SDRAM has an initial read delay of 70 ns and operates at a clock frequency of 100 MHz. This means that, once the memory is accessed, data will be available 7 clock cycles (cc) later (7 cc × 10 ns/cc = 70 ns). During that 7 clock cycle the processor can perform other activities, returning only when the data is available.

Although the same basic naming convention is used for both DRAM and SDRAM (RAS#, CAS#, WE# etc.), they do not have the same meaning. In an SDRAM these signals are interpreted as bits of an opcode, forming the SDRAM command set. These commands are listed below and are discussed in the section Theory of SDRAM Operation.

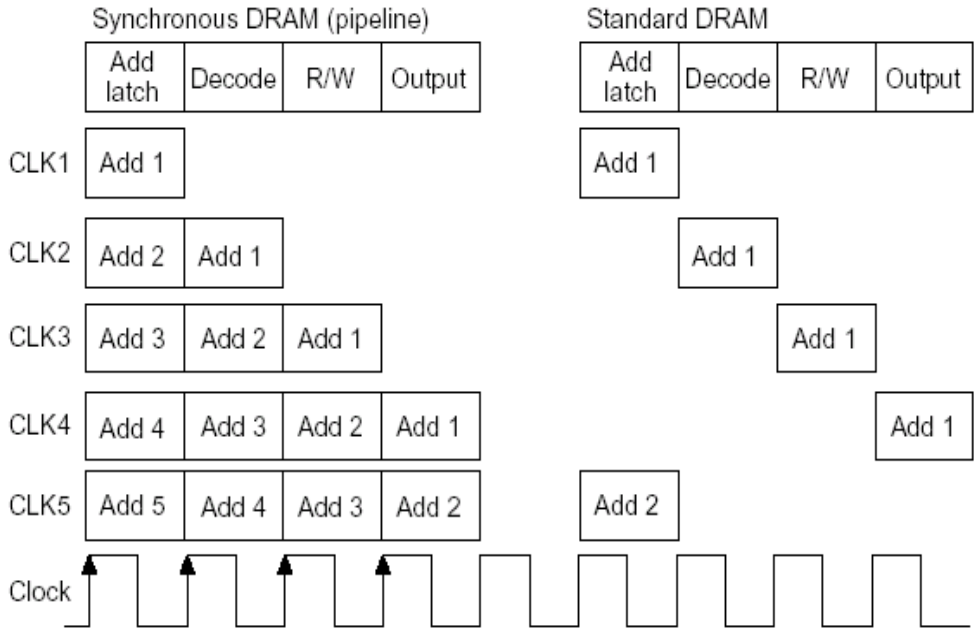


Figure 1-2 shows an example of SDRAM pipelining compared with a standard DRAM.

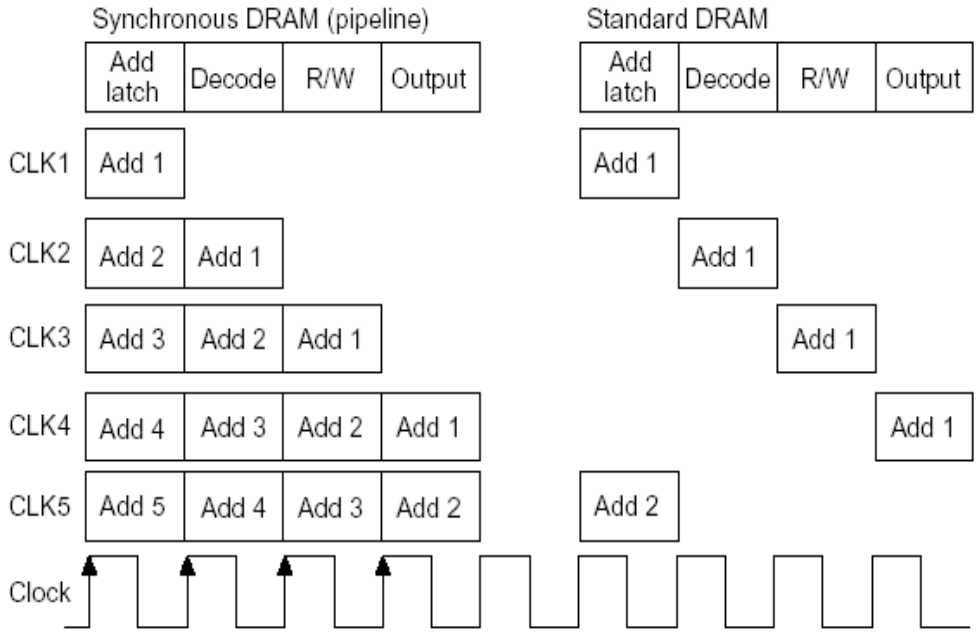


**Figure 1-2: SDRAM Pipelining compared with a standard DRAM**

## 1.4 Split-Bank Architecture

The internal memory array of the SDRAM consists of four separate banks. Which bank is selected depends on the state of the two highest order address bits. In a 128-Mbit SDRAM this would be A12 and A13. The four-bank scheme maximises bandwidth, since each bank can be accessed independently of the other. Once a given row in a given bank has been selected, it can be held active and accessed on every clock cycle by simply supplying a new column address. In addition, if data from four different rows is requested and the rows are in separate banks, much of the precharge can be hidden. One bank can be accessed while the others are being precharged. This approach mimics a four-way interleaved memory scheme. Once the row of each bank has been activated, alternate column accesses can occur every clock cycle, effectively causing the device to ping-pong between banks. This is because transition intervals are much faster between, rather than within, banks. In addition, multiple physical banks of devices are not needed, reducing the amount of space required and simplifying the memory-control logic.

## 1.5 Programmability

In order to retrieve memory data at the highest possible rate, the processor architectures implement burst mode. E.g. the processor can initiate a burst for a cache line fill. In a burst cycle, the processor generates only the first address of the read or write sequence, requiring the memory chip itself to generate the remaining addresses.

Burst mode reduces CPU bus usage since only one address needs to be generated rather than a specific address for every data location requested. The responsibility for determining the first

address and the size of the burst, as well as possibly generating the subsequent addresses, falls on the memory controller. If multiple burst lengths are supported the complexity of the memory controller is greatly increased.

SDRAMs have an on-chip burst counter which can be programmed based on the burst length requirements of a given cycle. The on-chip burst counter accepts the first address of the

burst cycle and increments in a particular sequence based on that address. The burst length is specified by programming the mode register after power-up. If multiple burst lengths are required in a system, the mode register can be reprogrammed prior to the execution of the new burst cycle. The device must be in the idle state each time a mode-register-set (MRS) operation is executed. No additional SDRAM control logic is required to support multiple burst lengths.

The SDRAM handles this internally based on information programmed into the mode register. The mode register contains the basic operating characteristics of the SDRAM.

Four burst lengths are supported: 2, 4, 8, and full-page. For example, a four-cycle burst means that the processor generates the first address and the SDRAM generates the remaining three addresses. The full-page option is very useful when refilling a cache after it has been flushed, and also for high-bandwidth graphics applications, where large amounts of data are required at one time, such as updating an on-screen image. The on-chip burst counter and the mode register greatly simplify SDRAM controller logic.

SDRAMs also support a programmable CAS# latency of either 1, 2 or 3. The CAS# latency of 1 is mentioned because of historical reason. Nowadays CAS# latency of 2 or 3 are state of the art. The CAS# latency determines the number of clock cycles which elapse between the time data is requested and the time it is transferred to the data outputs. The higher the number, the longer it takes to retrieve the first data item. However, higher clock rates demand more delay. Therefore, a CAS# latency of 3 can support faster clock rates. CAS# latency only applies to the first access of a data read.

Subsequent pipelined accesses to the same row can occur at the rate of one every clock. Figure 1-3 shows the burst lengths supported and the different CAS# latencies.

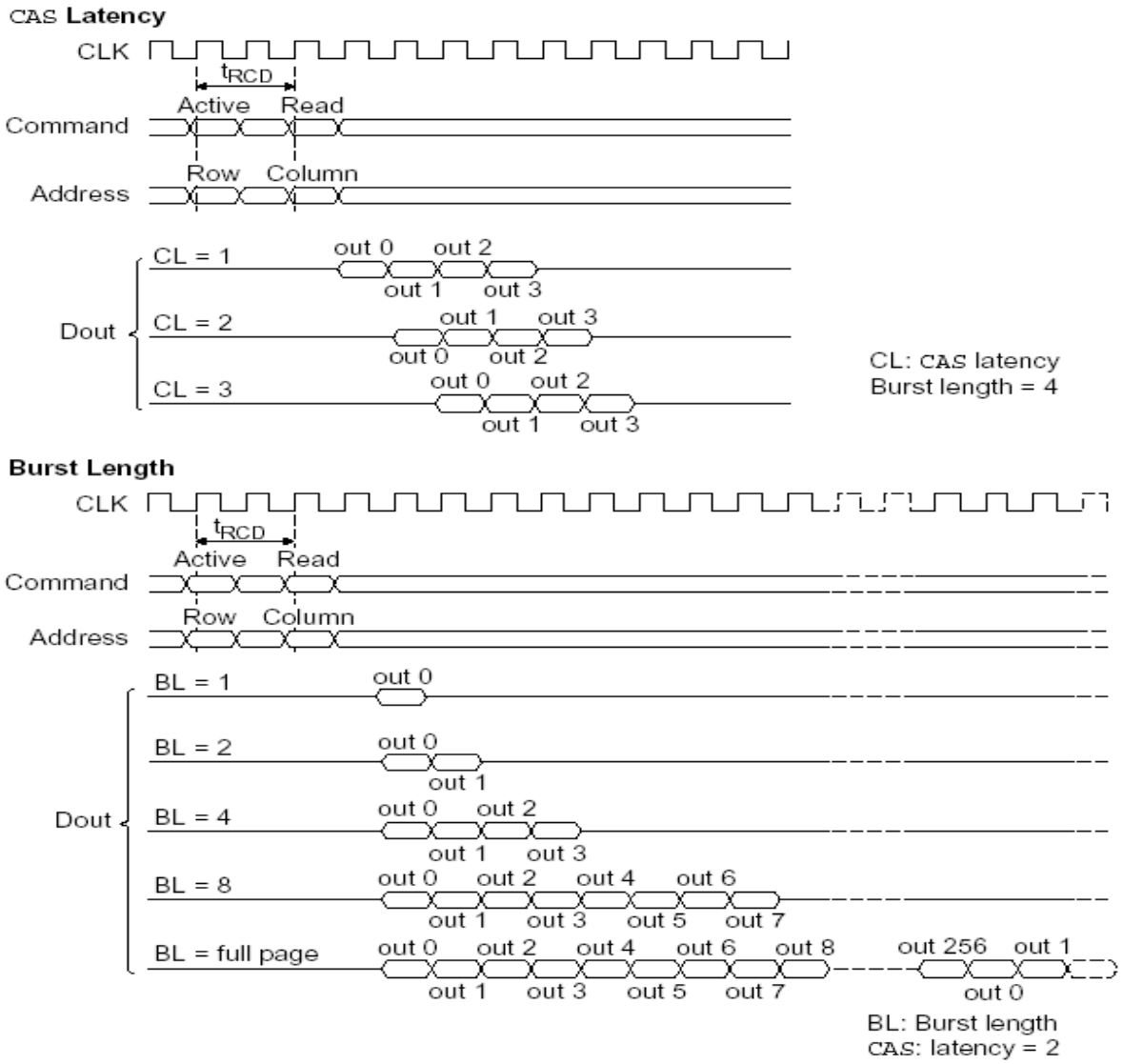


**Figure 1-3: SDRAM CAS Latency and Burst Length**

## 1.6 Individual Byte Enables

Typical SDRAMs are offered as 8 and 16-bit device that supports byte-selected writing

In the case of 8-bit devices DQM is used to support byte write operations. This signal performs the byte mask function. The signal controls one byte of data. DQM is active when D7–D0 are written.

In the case of 16-bit devices DQMU and DQML are used to support byte write operations. These signals perform the byte mask function. Each signal controls one byte of data. DQML controls when D7–D0 are written, and DQMU determines when D15–D8 are written. Without these

signals, a typical byte write would require a word (16-bit) read, then a byte mask, followed by a word write.

## 1.7 Two Refresh Modes

SDRAMs support two types of refresh mechanisms: auto-refresh and self-refresh. The state of the signal CKE determines which type is used.

For normal operation where the CKE signal is asserted and the clock is enabled (CKE=1), auto-refresh can be used. Auto refresh contains an on-chip refresh counter and is similar to a typical refresh cycle. No external addresses are required.

When the SDRAM is in low-power mode (CKE negated and the clock disabled), self-refresh mode can be used, allowing the SDRAM to refresh itself. In low-power mode the SDRAM requires only 4 mA of current in order to refresh itself.

## 1.8 Marking of SDRAMs

On top of each SDRAM package there should always be printed the partname followed by the production date and the three most important timing parameters CL, tRCD and tRP. CL defines the number of cycles between the latching of RAS and CAS command. The time tRCD is valid just for read access and defines the number of cycles between the CAS command is latched and the first data are bursted out of the SDRAM. And finally tRP defines the number of necessary precharge cycles. The timing parameters are printed on the chip and are expressed in cycle on, e.g. 2-2-2.

# 2 Theory of SDRAM Operation

## 2.1 Overview

This section explains the basic theory of operation of SDRAMs, focusing on those areas of the architecture unique to the SDRAM. Those areas of the architecture that are the same as a standard DRAM (cell structure, sense amp structure, etc.) are not covered.

The figure 2-1 shows a functional block diagram of a 128-Mbit Elpida SDRAM.

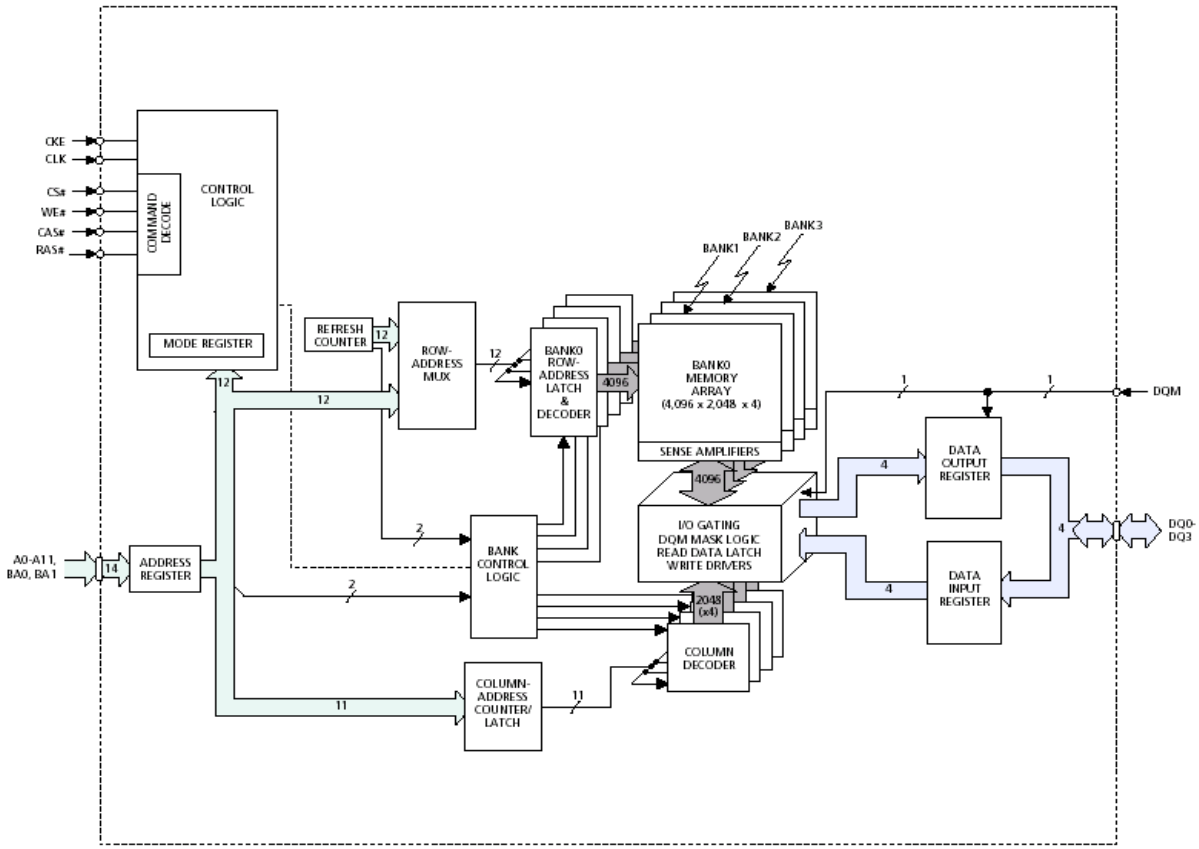


**Figure 2-1 Functional Block Diagram of a 128-Mbit SDRAM from Elpida**

The CS# input must be sampled low for either the row or column address to be latched. When the processor is not driving valid address and cycle information, CS# should be negated, so that all inputs are ignored. To minimise pin count, the SDRAM implements a multiplexed address bus. As with a standard DRAM, row address is presented to the input pins in one clock, and column address two or three clocks later. The time between the assertion of RAS# and the assertion of CAS# is defined as tRCD (RAS#-to-CAS# delay time). The address is latched and the input states are provided to the state machine. Because an SDRAM is pipelined, a given

cycle may or may not begin at the same time as its corresponding address is latched. The external signals of an SDRAM do not directly control the internal circuits.

Rather, the state of the inputs on any given clock causes transitions in the main state machine, which in turn determines the type of cycle to be run.

## 2.2 Power-On Sequence

The power-on sequence for the SDRAM products is defined by the JEDEC standards committee:

1. Apply power and start clock. Attempt to maintain NOP condition at the inputs.
2. Maintain stable power, stable clock, and NOP input conditions for a minimum of 200 microseconds.
3. Issue precharge command (PALL) for all banks of the device.
4. Issue 8 or more auto-refresh (REF) commands.
5. Issue mode-register-set (MRS) command to initialise the mode register.

The following figure 2-2 shows a timing diagram of the power-up sequence.

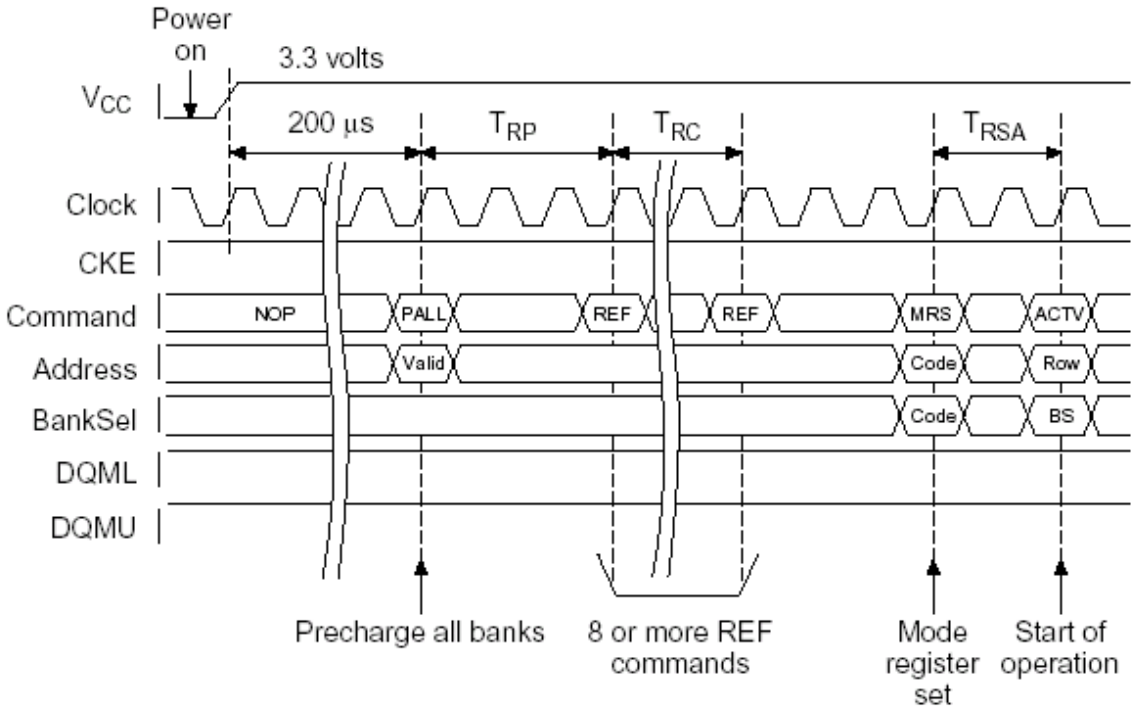


**Figure 2-2: Typical power-up sequence**

## 2.3 Command Operation

One significant difference between standard DRAMs and SDRAM architecture is the way in which internal accesses are executed. In standard DRAMs the toggling of the external pins has a direct effect on the internal memory array. In a SDRAM, the input signals are routed to a control logic block within the device which functions as the input to a state machine. Transitions in the state machine control the actual memory access. This approach allows pipelining of accesses to the SDRAM. For example, in a standard DRAM, once RAS# is asserted, the memory array cannot be accessed again until RAS# is negated and a fixed precharge time has elapsed. In addition, on same-page accesses, CAS# can only be asserted at intervals of two or more clocks, depending on the CAS# precharge time.

In a SDRAM, a second RAS# can be asserted within two clocks of the first RAS# as long as the access is to the opposite bank. Once RAS# is asserted for a given bank, CAS# can be asserted within two clocks, after which CAS# can be asserted every clock thereafter as long as the subsequent accesses are to the same row.

The control logic interprets transitions on the external pins as commands, which are in turn used to transition the state machine. Underneath table is the command truth table for a 128-Mbit SDRAM.

Note that the CKE input must be high for normal operation of the device. In addition, the CS# input must be asserted in order for any of the inputs to be recognised.

If CS# is sampled high during any rising clock edge, the information on the external pins is ignored.

After the device is powered-up the mode register can be programmed. The mode register is undefined on power-up. Therefore, cycle behaviour is also undefined until this register is programmed, since the contents of the mode register determines the characteristic of each cycle. In addition to power-up, the mode register can be programmed during normal operation whenever the state machine is in idle state, in order to change the operating characteristic of the device. The mode register, Figure 2-3, is indicated by the MRS command and is listed in the table shown in section mode register.

The mode register is programmed by driving certain values onto the address pins while the main control signals are all driven low as shown. In the SDRAM architecture the assertion of RAS# is completely separate from the assertion of CAS# and has its own command. Whenever a cycle is initiated from the idle state, the first command to be executed is the ACTV command. With the exception of precharge and refresh operations, RAS# is only asserted during the ACTV command. At the time the ACTV command is executed the SDRAM has no idea what type of data operation is to be performed. Its only function is to prepare the given row for a memory access. The assertion of CAS# causes the various read or write commands to be executed, which typically occur two cycles after RAS# assertion. The following table 2-1 lists the SDRAM commands.

| Command | Symbol | CKE n - 1 | CKE n | CS | RAS | CAS | WE | A12/A13 | A10 | A0 to A11 |
|---|---|---|---|---|---|---|---|---|---|---|
| Ignore command | DESL | H | × | H | × | × | × | × | × | × |
| No operation | NOP | H | × | L | H | H | H | × | × | × |
| Burst stop in full page | BST | H | × | L | H | H | L | × | × | × |
| Column address and read command | READ | H | × | L | H | L | H | V | L | V |
| Read with auto-precharge | READ A | H | × | L | H | L | H | V | H | V |
| Column address and write command | WRIT | H | × | L | H | L | L | V | L | V |
| Write with auto-precharge | WRIT A | H | × | L | H | L | L | V | H | V |
| Row address strobe and bank active | ACTV | H | × | L | L | H | H | V | V | V |
| Precharge select bank | PRE | H | × | L | L | H | L | V | L | × |
| Precharge all bank | PALL | H | × | L | L | H | L | × | H | × |
| Refresh | REF/SELF | H | V | L | L | L | H | × | × | × |
| Mode register set | MRS | H | × | L | L | L | L | V | V | V |

Note: H: $V_{IH}$. L: $V_{IL}$. ×: $V_{IH}$ or $V_{IL}$. V: Valid address input

**Table 2-1 SDRAM Command List**

**DESL:** DESL is the default command executed by the state machine whenever CS# is sampled high. The state of the external pins is ignored. Execution of this command causes the state machine to remain in its current state.

**NOP:** No Operation is similar to DESL except that CS# is asserted, causing the control logic to sample each of the external pins. If all the control signals are negated as shown in table 2-1, a NOP command is decoded. The command has no effect on the internal operation of the state machine and any cycles in progress are allowed to continue.

**BST:** The BST command can only be executed during a full-page burst cycle. Execution of this command is illegal during execution of any other burst operation. Execution of the BST command causes the output buffers to become high-impedance. The full-page burst is useful for high-speed graphics applications such as reading video from memory and mapping it to the screen. If the system is designed so that not all of the full page burst transfers are required to complete the screen update operation, the burst can be terminated as necessary by execution of the BST command.

A full-page burst that has been terminated cannot be restarted where it left off. Rather, the burst counter is reset. Therefore, if the full-page burst mode is used for a normal CPU operation such as refilling a cache, and the BST command is executed during the burst, there is no way to restart the burst. Either the entire burst cycle can be re-executed, or smaller bursts to those locations not yet accessed can be executed to retrieve the entire full-page burst data. For this option the mode register must be reprogrammed with a burst-length value other than full-page before subsequent

burst cycles can be executed. The BST command is required to terminate the full-page burst even if the burst is allowed to complete. Failure to execute the BST command at the end of a completed burst causes the burst counter to wrap-around to its starting address. The same data is again output until the BST command is executed.

**READ:** The read command specifies the column address corresponding to the row address executed by the previous ACTV command. The READ command is optimised for single read accesses.

When the READ command is executed, data is available a set number of clocks later as defined by the CAS# latency mode in the mode register. Latencies of 2 and 3 cycles are supported. Another READ or READA command can be executed while the current READ command is in progress, causing the state machine to transition so that a new row address can be processed. However, execution of either the WRIT or WRITA commands during execution of a READ command causes termination of the read operation and allows the write operation to begin. In addition, execution of either the PRE or PALL commands during execution of a READ command also causes termination of the read operation and allows a precharge operation to begin. After completion of a READ command, the output buffers are high impedance.

If the CKE signal is driven low during execution of the READ command, the data is held and continues to be output until CKE is driven high.

**READA:** The READA command is executed when it is necessary to read the memory array and at the same time perform a precharge operation. This command only works when the burst length size is programmed to 2, 4, or 8. Single-transfer read cycles and full-page bursts cannot be executed using the READA command. Execution of the READA command allows the active bank to be precharged prior to completion of the read operation. Overlapping the actual access with the precharge operation helps hide the latency required while the data flows through the sense amplifiers to the output pins and increases the overall throughput for that bank. Unlike the READ command, the READA command is not terminated by execution of the WRIT, WRITA, PRE, or PALL commands while the read-with-auto-precharge operation is in progress. If the CKE signal is driven low during execution of the READA command, the data is held and continues to be output until CKE is driven high.

**WRIT:** The write command specifies the column address corresponding to the row address executed by the previous ACTV command. Write cycles can be either single transfer or burst transfer. Multiple back-to-back burst accesses to different rows of the same bank are better facilitated using the WRITA command. If the CKE signal is driven low during execution of the WRIT command, the data is held in the input latches and all input pins are ignored until CKE is driven high.

Execution of either a READ or READA command while the write operation is in progress causes the write operation to be terminated and the read operation to begin. In addition, execution of either a PRE or PALL command while the write operation is in progress causes the write operation to be terminated and a precharge operation to begin. Unlike the READ command, where a subsequent READ command is allowed as long as it is to the opposite bank, execution of a subsequent WRIT or WRITA command while the write operation is in progress causes the operation to be terminated and a new write operation to begin.

**WRITA:** The WRITA command is executed when it is necessary to write the memory array and at the same time perform a precharge operation. This command only works when the burst length size is programmed to 2, 4, or 8. Single-transfer write cycles and full-page bursts cannot be executed using the WRITA command. Execution of the WRITA command allows the active bank to be precharged prior to completion of the write operation. Overlapping the actual access with the precharge operation helps hide the latency required while the data flows from the input pins through the sense amplifiers to the array and increases the overall throughput for that bank.

Unlike the WRIT command which can be easily terminated, the WRITA command cannot be interrupted. Execution of the READ, READA, WRIT, WRITA, PRE, or PALL commands while the write-with-auto-precharge operation is in progress are illegal. If the CKE signal is driven low during execution of the WRITA command, the data is held in the input latches and all input pins are ignored until CKE is driven high.

**ACTV:** The ACTV command is executed any time a read or write operation is to be performed. The ACTV command is responsible for latching and decoding the row address and activating the appropriate row in the array. The ACTV command is unaware of the type of operation to be performed, and does not require this information in order to complete. Once the ACTV command is complete a read, write, or precharge operation can be performed and the corresponding column address driven. A subsequent ACTV command is allowed while the current ACTV command is in progress, as long as the access is to the opposite bank.

In most SDRAM devices there are more rows as columns in the memory array. Therefore, additional address bits are required to adequately decode the row address relative to the column address. For example, in a 128-Mbit SDRAM with a 16-bit data bus, twelve address bits (A11–A0) are required to decode one of 4k rows, while only nine address bits (A8–A0) are required to decode one of 512 columns. During the ACTV command the address bits A12 and A13 are used as a bank select, while bits A11–A0 determine the row address. During column-address generation address bits A12 and A13 is also used as a bank select, but only address bits A8–A0 are required for the column address. On read cycles, address bit A10 is used to determine whether the READ or READA command is executed. On write cycles, address bit A10 is used to determine whether the WRIT or WRITA command is executed.

**PRE:** The PRE and PALL commands both perform the precharge operations: the PRE command precharges a single bank, while PALL precharges both banks. As shown in table 2-1, the simultaneous assertion of RAS# and WE# during normal execution denotes a precharge operation. During normal operation, address bit A12 and A13 is used to select which bank is precharged. Address bit A10 indicates whether the PRE or PALL operation is executed. Subsequent cycles cannot be executed while the PRE command is being executed and the precharge operation is in progress. After the operation is completed the state machine switches to the idle state. Only then the ACTV for a follow-on cycle can be executed.

**PALL:** Execution of the PALL command causes control logic to initiate a precharge operation to both banks of the SDRAM simultaneously. During a PALL operation address bit A12/13 are ignored. Address bit A12 and 13 indicates whether the PRE or PALL operation is executed. Subsequent cycles cannot be executed while the PALL command is being executed and the

precharge operation is in progress. After the operation is completed the state machine switches to the idle state. Only then can the ACTV for a follow-on cycle be executed.

**REF:** The SDRAM architecture supports two types of refresh modes, which are defined by the REF and SELF commands. Only the REF command can be executed during normal operation. The SELF command is restricted to refresh operations in low-power (clock- suspend-mode) mode. The REF command can only be entered when the state machine is in the idle state; that is, refresh operations cannot commence until all current cycles have completed.

Each SDRAM row require a refresh cycles every 64 ms.

Internally, execution of the REF command performs the same function as a CAS-before-RAS refresh used in standard DRAMs.

Refresh addresses and bank selects are provided by an on-chip refresh counter. Therefore, no external address is required as indicated in table 2-1 by an X (don't care) representing the address bus. In addition, each bank is precharged after a refresh operation. Hence the PRE and PALL commands need not be executed after a refresh operation. The REF command cannot be interrupted. A refresh operation is always allowed to complete, after which the state machine returns to the idle state.

| | | CKE | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Current state | Command | n - 1 | n | $\overline{CS}$ | $\overline{RAS}$ | $\overline{CAS}$ | $\overline{WE}$ | Address |
| Active | Clock suspend mode entry | H | L | H | × | × | × | × |
| Any | Clock suspend | L | L | × | × | × | × | × |
| Clock suspend | Clock suspend mode exit | L | H | × | × | × | × | × |
| Idle | Auto-refresh command (REF) | H | H | L | L | L | H | × |
| Idle | Self-refresh entry (SELF) | H | L | L | L | L | H | × |
| Idle | Power down entry | H | L | L | H | H | H | × |
| | | H | L | H | × | × | × | × |
| Self refresh | Self refresh exit (SELFX) | L | H | L | H | H | H | × |
| | | L | H | H | × | × | × | × |
| Power down | Power down exit | L | H | L | H | H | H | × |
| | | L | H | H | × | × | × | × |

Note: H: $V_{IH}$. L: $V_{IL}$. ×: $V_{IH}$ or $V_{IL}$.

**Table 2-2: CKE Truth Table, 128-Mbit SDRAM**

**SELF:** Like the REF command, the SELF command can only be executed when the state machine is in the idle state. Negation of the CKE signal indicates that the clock has been suspended. This low-power mode can be used when the SDRAM will be idle for long periods of

time and is ideal for battery-powered devices. The SELF command is necessary since during low-power mode the clock is suspended, hence the REF command cannot be issued. Low-power mode greatly reduces the amount of current usage. During normal operation a typical SDRAM can use 120 mA. However, during low-power mode where the device refreshes itself, only 4 mA of current are consumed. The SELF command continues execution as long as the CKE signal is low.

**SELFX:** Transition of the CKE signal from low to high causes execution of the SELFX command, terminating the self-refresh operation. As shown in table 2-2, there are two conditions under which the SELFX command is executed. If CKE switches from low to high and the CS# signal is also high, the SELFX command is executed regardless of the state of the remaining pins, and the state machine returns to the idle state. If CS# is low at the time of the CKE transition, the other control signals must be sampled in the states shown in order for the SELFX command to execute. All other commands are illegal at this time. The transition of CKE from low to high can occur at any time, leaving the refresh counter in an undefined state. Therefore, when exiting from low-power mode, 4096 cycles of auto-refresh (REF) are required to update the refresh counter.

**PWRDN:** The PWRDN command can be executed when the state machine is in the IDLE state, causing the device to suspend all operation and enter power-down mode. The PWRDN command is executed under one of two conditions. If the CKE signal switches from high to low, and CS# is high, PWRDN is executed. If the CKE signal switches from high to low, and CS# is low, other inputs are also sampled and must be in the correct states as shown in table 2-2 in order for PWRDN to execute. While in power-down mode all input circuits to the array are high-impedance. No self-refresh cycles are performed while in power-down mode.

**PWRDNX:** The device can remain in power-down mode for an indefinite amount of time.

There are two conditions under which the PWRDNX command is executed; both require a transition from low to high on the CKE signal:

- Sampling the CS# signal high during the CKE transition causes execution of the PWRDNX command.

- If the CS# signal is sampled low during the CKE transition, additional inputs must be in the correct state as shown in table 2-2 in order for the PWRDNX command to execute. Execution of the PWRDNX command causes the machine to return to the IDLE state.

**MRS:** The MRS command is executed on power-up or whenever the controller needs to change the operating parameters of the device. The MRS command initialises the mode register. The mode register maintains the operating characteristics of the device, such as burst length size, CAS# latency, write mode, etc. The mode register is defined at the beginning of the command operation (figure 2-3).

## 2.4 Precharge Operations

Precharge operations can be performed on both read and write cycles. The operation in turns off the word-line in the selected array to store full-level data in the memory cells. Then, it

precharges the bit-lines to VDD/2 by shorting them together in preparation for the next active cycle.

There are three precharge options:

• Precharge a single bank.

• Precharge all banks simultaneously.

• Execute a read or write command with auto-precharge.

Execution of the PRE command precharges a single bank as determined by the state of the highest-order address bit. When precharging a single bank, the state of the second-highest-order address bit must be low. The precharged bank is selected via the bank select bits, A12 and 13, Execution of the PALL command precharges all banks simultaneously. For this command the state of the second-highest order address bit must be high. The state of the highest-order address bit is irrelevant since all banks are to be precharged. A PALL command is the first instruction that should be executed on power-up. This allows the internal SDRAM state machine to precharge all banks, then move automatically to the idle state where an MRS command can be executed. Execution of either the READA or WRITA command executes the appropriate cycle along with a precharge to the same bank at the end of the operation. Because the precharge operation consumes some number of cycles, the READA and WRITA commands perform the precharge during the actual cycle, effectively hiding some of the precharge operation latency. The number of cycles required to perform a precharge is defined in the AC characteristics section of the device datasheet.

## 2.5 Mode Register

The mode register is unique to the SDRAM architecture and is a key feature that distinguishes it from the standard DRAM architecture. The mode register contains the parameters under which the device operates. The size of the mode register is equivalent to the number of address pins on the device and is written during a Mode Register Set (MRS) cycle, which is generated whenever all the control inputs are low. Valid address is driven onto A13–A0 and the mode register is written. The mode register must be reprogrammed whenever any of the parameters change. Therefore, systems that require multiple burst-length sizes, such as those having different data and instruction cache line widths, may require an MRS cycle each time the burst length changes. There are two ways to deal with this from a design standpoint:

- The controller can manage the mode register so that, when a memory cycle occurs where the burst length is different from that of the previous cycle, the controller can insert an MRS cycle and reprogram the mode register accordingly.

- The controller set the burst length at power-up and leave it. For processor cycles that require more data than the set burst length, the controller can perform as many bursts as it takes to supply the processor with the correct amount of data. For example, assume the burst length is set to two transfers and the processor requires a four-transfer burst. The controller can perform a two-transfer burst, then another. This approach would require the controller to

generate the first address for the second burst. During single transfer cycles the controller can simply ignore data from the second transfer.

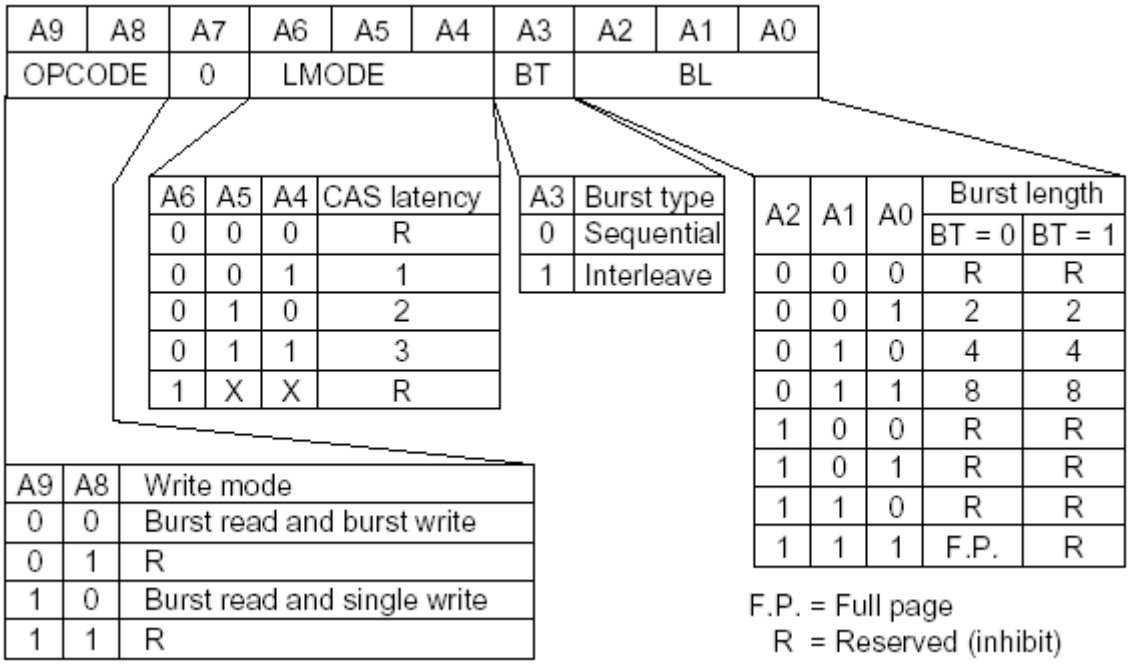
A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
OPCODE 0 LMODE BT BL
A6 A5 A4 CAS latency
0 0 0 R
0 0 1 1
0 1 0 2
0 1 1 3
1 X X R
A3 Burst type
0 Sequential
1 Interleave
A2 A1 A0 Burst length
BT = 0 BT = 1
0 0 0 R R
0 0 1 2 2
0 1 0 4 4
0 1 1 8 8
1 0 0 R R
1 0 1 R R
1 1 0 R R
1 1 1 F.P. R
A9 A8 Write mode
0 0 Burst read and burst write
0 1 R
1 0 Burst read and single write
1 1 R
F.P. = Full page
R = Reserved (inhibit)

**Figure 2-3: Mode Register, 128-Mbit SDRAM**

Regardless of the size of the burst cycle requested, the SDRAM always transfers the number of data items as defined in the mode register. For example, if the mode register is set to a burst length size of four, and the processor requires only a single data transfer, the SDRAM still transfers four data items. It is up to the controller to assure that the remaining three data transfers are ignored by the processor. Mode register bits 2–0 define the burst length size. Burst lengths of 2, 4, 8, and full-page are supported. Those entries marked ‘R’ are reserved and equate to an illegal opcode, which if used can cause unpredictable device behavior. A full-page burst length size can only occur when sequential burst mode is used.

**Bit 3:** Bit 3 defines the burst type:

- Sequential burst mode means that each transfer of a given burst resides at a sequential address. The burst counter advances sequentially and the data is returned. The counter simply starts at the address defined by the processor and wraps around until all of the data is transferred.

- Interleave burst mode (not supportet by SH4) allows support of the Intel sub-block ordering protocol, where one of four burst sequences can occur based on the value of the starting address. Interleave mode affects the way address bits A0 and A1 are incremented in the SDRAM controller. Table 2-3 shows each of these sequences.

Burst length = 2

| Starting Ad. | Addressing(decimal) | |
|---|---|---|
| A0 | Sequential | Interleave |
| 0 | 0, 1, | 0, 1, |
| 1 | 1, 0, | 1, 0, |

Burst length = 4

| Starting Ad. | | Addressing(decimal) | |
|---|---|---|---|
| A1 | A0 | Sequential | Interleave |
| 0 | 0 | 0, 1, 2, 3, | 0, 1, 2, 3, |
| 0 | 1 | 1, 2, 3, 0, | 1, 0, 3, 2, |
| 1 | 0 | 2, 3, 0, 1, | 2, 3, 0, 1, |
| 1 | 1 | 3, 0, 1, 2, | 3, 2, 1, 0, |

Burst length = 8

| Starting Ad. | | | Addressing(decimal) | |
|---|---|---|---|---|
| A2 | A1 | A0 | Sequential | Interleave |
| 0 | 0 | 0 | 0, 1, 2, 3, 4, 5, 6, 7, | 0, 1, 2, 3, 4, 5, 6, 7, |
| 0 | 0 | 1 | 1, 2, 3, 4, 5, 6, 7, 0, | 1, 0, 3, 2, 5, 4, 7, 6, |
| 0 | 1 | 0 | 2, 3, 4, 5, 6, 7, 0, 1, | 2, 3, 0, 1, 6, 7, 4, 5, |
| 0 | 1 | 1 | 3, 4, 5, 6, 7, 0, 1, 2, | 3, 2, 1, 0, 7, 6, 5, 4, |
| 1 | 0 | 0 | 4, 5, 6, 7, 0, 1, 2, 3, | 4, 5, 6, 7, 0, 1, 2, 3, |
| 1 | 0 | 1 | 5, 6, 7, 0, 1, 2, 3, 4, | 5, 4, 7, 6, 1, 0, 3, 2, |
| 1 | 1 | 0 | 6, 7, 0, 1, 2, 3, 4, 5, | 6, 7, 4, 5, 2, 3, 0, 1, |
| 1 | 1 | 1 | 7, 0, 1, 2, 3, 4, 5, 6, | 7, 6, 5, 4, 3, 2, 1, 0, |

**Table 2-3: Sequential and interleave burst**

**Bits 6–4:** Bits 6–4 define the CAS# latency on read cycles, which can be 1, 2, or 3. CAS latency one is mentioned because of historical reasons. The latest SDRAMs do not support CAS latencies of 1. The remaining entries in the latency mode are undefined. The CAS# latency determines the number of clocks that elapse between the time the column is accessed and the time the data appears on the output pins. A latency of 2 would mean that the column would be accessed and the data made available two clocks later. Conversely, a CAS# latency of 3 means that data is not available until three clocks after the column is accessed. Figure 2-4 shows how data output is affected by each CAS# latency.

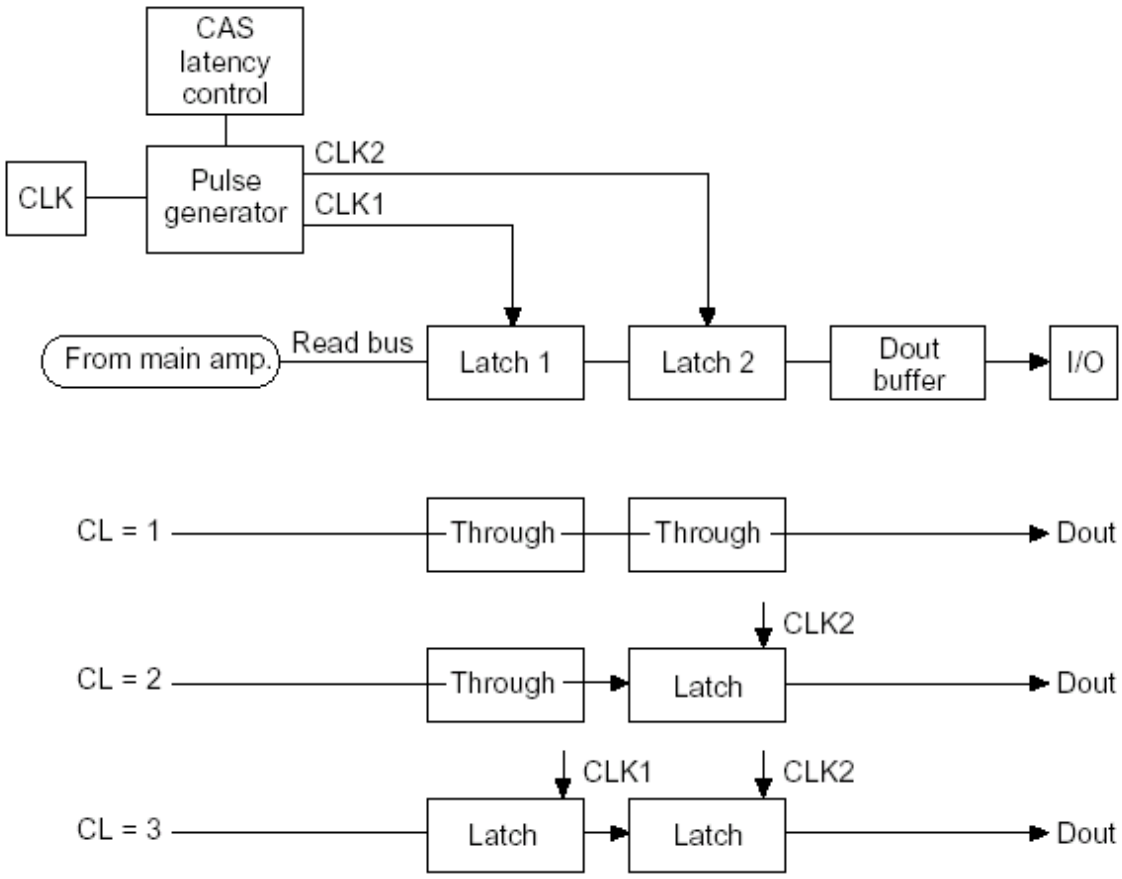


**Figure 2-4: Output effected on CAS Latency**

**Bit 7:** Keep this bit Low at the mode register set cycle. If this pin is high, the vender test mode is set.

**Bits 13…8:** Bits 13 … 8 define the write mode. SDRAMs offer two types of write modes: a burst read can either be followed by a single write, or a burst write. The amount of data transferred on a burst write is defined by mode register bits 2–0. If the processor generates a burst read followed by a single write cycle and the mode register is programmed in burst write mode (A13 … A8 = 0), the controller must reprogram the mode register before allowing the write cycle to complete. This is because the burst write is represented by only one assertion of the WE# signal for one clock, as opposed to four separate WE# assertions as would occur in a standard DRAM. A burst read followed by a single write can be useful in graphics applications. For example, a burst read can be used to refresh the video screen, and single writes can be used to update specific portions of the screen. This mode can also be used to support a CPU with a write-through primary cache, where each write to the primary cache is also driven onto the external bus.

## 2.6 ACTV Command Sequence

The simultaneous assertion of RAS# and CS# causes execution of the ACTV command. The ACTV command is executed on both read and write operations and performs the following functions:

1. Sets the row address as determined by the state of the address inputs.

2. Latches the row address and bank select signals. The appropriate bank select is determined by the state of the highest order address bit.

3. Decodes the row address. The row address is first multiplexed with the refresh address counter, after which it enters the row address decoder of the enabled bank. In a 128-Mbit SDRAM, the row address decoder is responsible for taking the 14-bit input and determining which of the 4k rows in one of the four banks is activated. The lower bits are used to activate the actual row (A0-A11), while the address bits A12 and A13 determines the bank. The state of the address bits/ bank select bits causes only one bank to be activated, thereby reducing power consumption.

4. Activates the appropriate line driver based on the row address decode.

The next figure shows a flow diagram of the ACTV command. Refer to section 2.3, Command Operation, for more information on the ACTV command.

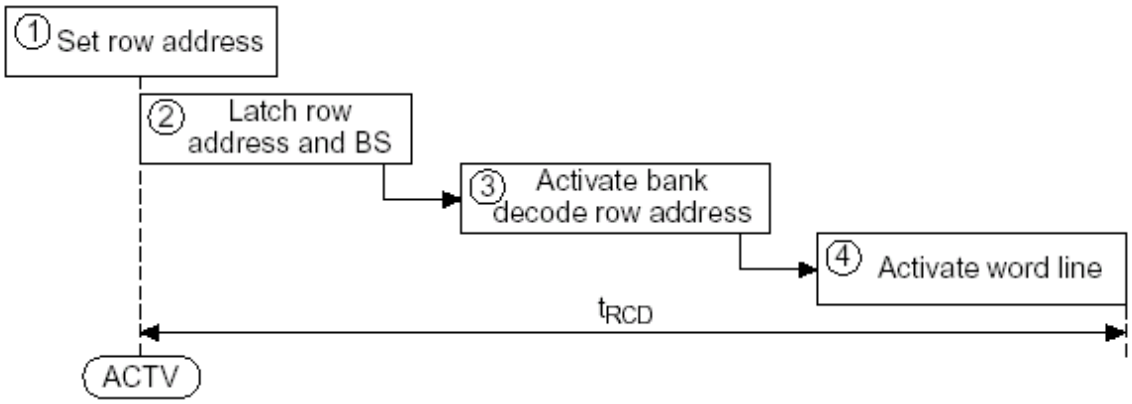


**Figure 2-5: ACT Command , Flow Chart**

## 2.7 Read Operation

All data cycles start with execution of the ACTV command, which is generated by the state machine whenever CS# and RAS# along with valid address are asserted simultaneously. The SDRAM commands are defined in table 2-1. After execution of the ACTV command the array still does not know if the pending cycle will be a read, a read with auto-precharge, a write, or a

write with auto-precharge. Which subsequent command is executed depends on the state of address bit A10 as well as the WE# signal. These parameters are defined along with the assertion of CAS#.

After the ACTV command is executed, a basic read operation begins by the assertion of CAS# and CS#, along with valid address. The WE# signal is inactive at this time. A low in address bit A10 indicates a read command (READ). A high on address bit A10 indicates a read with-auto-precharge (READA).

The read operation is responsible for outputting data onto the data bus after the ACTV command has been executed. Figure 2-6 shows a block diagram of a read operation. The numbers shown throughout the diagram correspond to the numbered sequence explained below.

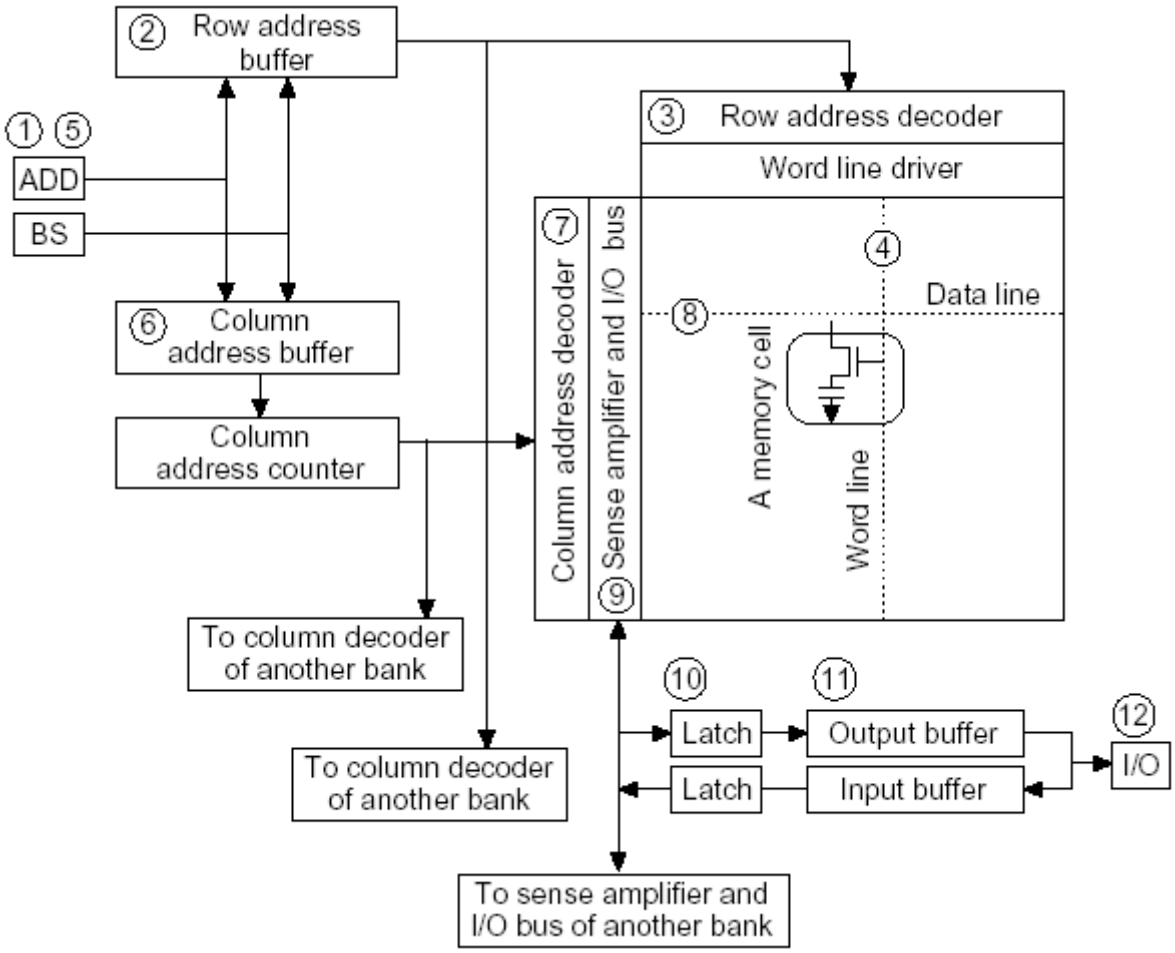


**Figure 2-6: Read Operation, Block Diagram**

**Read Command Sequence:** On a read cycle, the ACTV command is responsible for performing steps 1 through 4 as explained in section 2.6. The column address is latched three

clocks after the row address. The assertion of CAS# and CS# along with valid address causes the state machine to transition to the READ state. Steps 5 through 12 relate to the execution of the READ command and correspond to the numbered sequence in figures 2-6 and 2-7.

5. The column address is set based on the state of address inputs A8–A0. Address bit A12 and A13 are the bank select. Address bit A10 determines whether the command is a standard read command (READ), or a read with auto-precharge (READA). During read cycles, SDRAMs can accept a new column address every clock as long as the row is active.

6. The column address is latched and routed through the burst counter to the column decoder. Whether or not the burst counter is activated depends on the state of the “burst length size” entry in the mode register. When the burst length is 2, 4, or 8, the data output buffer shown in step 11 of figure 2-6 automatically switches to the high impedance state in the next cycle after the last burst data has been transferred. When the burst length is full-page and the last data has been transferred, the wrap-around nature of the burst counter causes the data to be repeatedly output until the burst stop (BST) command is executed.

7. The address enters the column address decoder. The decoder determines which one of the 512 columns is activated.

8. The correct data line is activated and the data retrieved from the memory array.

9. Data is routed through the sense amplifiers and internal I/O bus.

10. The data enters the CAS# latency control latch where it is latched the appropriate number of cycles based on the CAS# latency setting programmed into the mode register. This can be between two and three clock cycles.

11. Then the data enters the output buffers where it remains until the data mask signals (DQML/DQMH) are asserted.

12. The data to be driven onto the output pins.

Figure 2-7 shows a flow chart of a READ operation. The numbers shown correspond to those in the block diagram in figure 2-7.

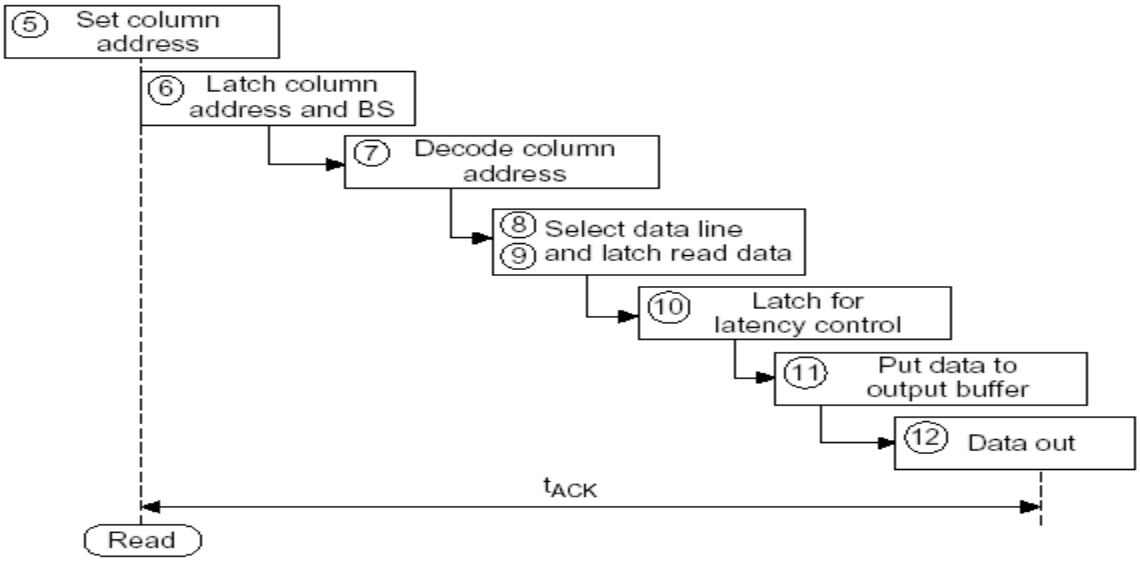


**Figure 2-7: READ Operation, Flow Chart**

## 2.8 Write Operation

As mentioned in the section 2.7, Read Operation, all data cycles start with execution of the ACTV command, which is generated by the state machine whenever CS# and RAS# along with a valid address are asserted simultaneously. There are two types of read-followed-by-write operations as defined by bits 13–8 of the mode register. One is a burst read followed by a burst write, where the burst length size is defined by bits 2–0 of the mode register. The other is a burst read followed by a single write, where only the burst read length size is defined in the mode register. The subsequent write is always a single write, effectively bypassing the burst length size of the mode register. In this mode, if a multiple transfer write cycle is generated by the processor, such as in a burst write back of the primary cache, one of two actions can be taken. Upon decoding the incoming cycle, the controller can

• Reprogram the mode register to accept a burst write, perform the cycle, and then again reprogram the mode register to accept only single cycle writes, or

• Accept the burst write and generate four single write cycles to the SDRAM. Although the mode register need not be reprogrammed, it is necessary for the controller to generate subsequent addresses to the SDRAM. In addition, the controller must be prepared to increment memory addresses either in sequential fashion, or conform to the interleaved subblock ordering protocol (not supported by SH-4). In this case reprogramming the mode register is the more efficient solution.

A basic write operation begins by the assertion of CAS# and CS#, along with valid address.

The WE# signal is also asserted at this time. A low on address bit A10 indicates a write (WRIT) command, while a high indicates a write-with-auto-precharge (WRITA).

The write operation is responsible for accepting data from the data bus and writing it to the memory array after the ACTV command has been executed. Figure 2-8 is a block diagram of a write operation. The numbers shown throughout the diagram correspond to the numbered sequence explained below.

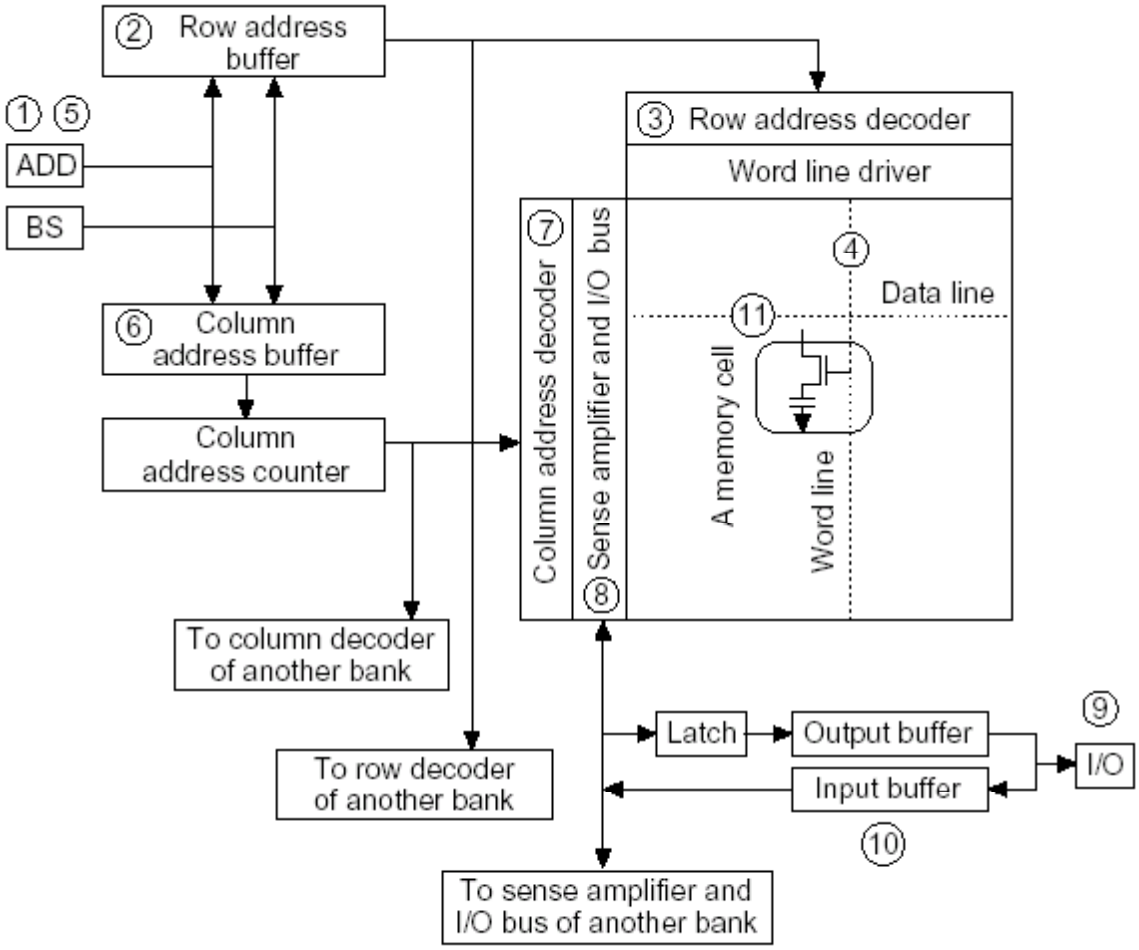


**Figure 2-8: Write Operation, Block Diagram**

The simultaneous assertion of RAS# and CS# causes execution of the ACTV command.

Steps 1–4 in figure 2-8 represent execution of the ACTV command and are explained in the ACTV Command Sequence section.

**Write Command Sequence:** The column address is latched three clocks after the row address. The assertion of CAS# and CS# along with valid address causes the state machine to transition to the WRITE state. Steps 5–11 relate to the execution of the WRIT command. This numbered

sequence corresponds to that in figures 2-8 and 2-9. The physical location in the SDRAM where each step is performed is shown in figure 2-8. Figure 2-9 is a flow chart of the WRIT command.

5. The column address is set based on the state of address inputs A8–0. Address bits A12 and A13 are used for the bank select. Address bit A10 determines whether the command executed is a write (WRIT) or a write-with-auto-precharge (WRITA). On write cycles, SDRAMs can accept a new column address on every clock as long as the row is active.

Simultaneous with the column address being driven, the SDRAM accepts the write data which enters the on-chip input buffer.

6. The column address is latched and routed through the burst counter to the column decoder. If the operation is a read-followed-by-write, whether or not the burst counter is activated depends on the state of the “burst length size” and “opcode” entries in the mode register. The write data is latched at this time.

7. The address enters the column address decoder. The decoder determines which column is activated.

8. The correct data line is activated based on the column address decode.

9. Step 9 occurs at the same time as step 5. The SDRAM accepts the write data which enters the on-chip input buffer.

10. Step 10 occurs at the same time as step 6. The write data is latched internally at this time.

11. The write data is written to the memory cell.

Figure 2-9 shows a flow chart of a WRITE operation. The numbers shown correspond to those in the block diagram in figure 2-8.

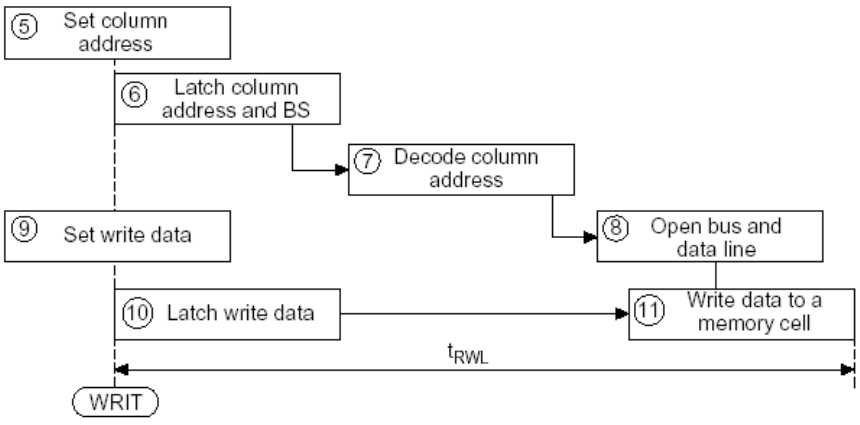
⑤ Set column address
⑥ Latch column address and BS
⑦ Decode column address
⑧ Open bus and data line
⑨ Set write data
⑩ Latch write data
⑪ Write data to a memory cell
$t_{RWL}$
WRIT

**Figure 2-9: Write Operation, Flow Chart**

## 2.9 State Machine

Figure 2-10 is a simplified diagram of the main state machine. Transitions between states are defined by the commands listed in the section 2.3, Command Operations.

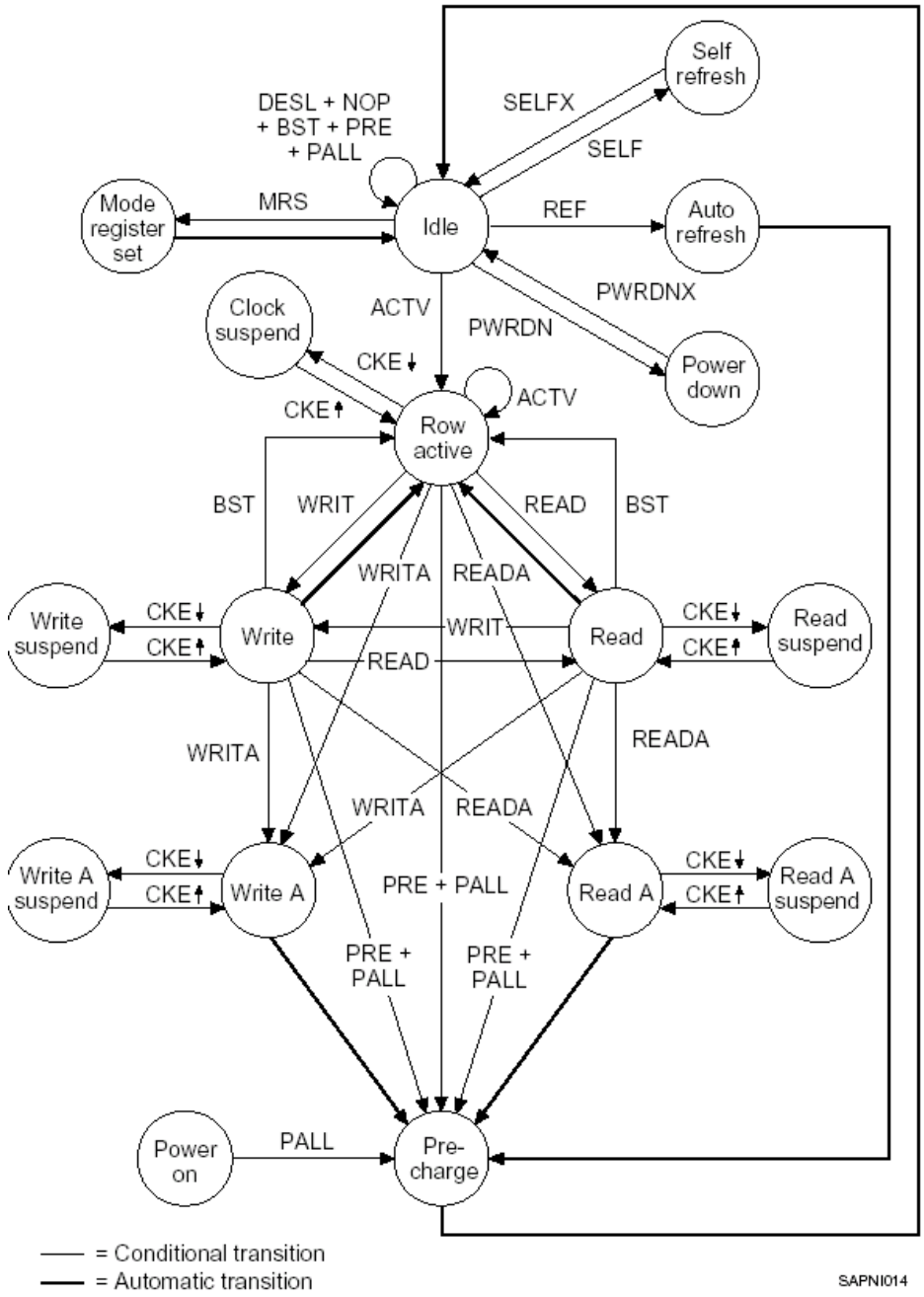


**Figure 2-10 State Diagram**

After the device is powered-up and the mode register is programmed, PALL command is executed which causes the machine to transition to the PRECHARGE state where a precharge operation is performed on all internal banks. Once the precharge operation is completed the machine automatically transitions to the IDLE state.

Refresh, power-down, and mode register operations can be performed while the machine is in the IDLE state.

Execution of the SELF command causes the machine to transition to the SELF-REFRESH state. It is necessary for CKE# to transition from high to low in order to enter the SELF-REFRESH state. While in the SELF-REFRESH state the memory array is in low-power mode. An internal refresh counter allows data integrity to be maintained.

Execution of the SELFX command, which occurs when the CKE signal switches from low to high, causes the machine to return to the IDLE state. As stated in section 2.3, Command Operation, low-power mode can be used during long-latency I/O operations. Low-power mode is also very useful in battery operated devices.

The AUTO-REFRESH state is entered by execution of the REF command, which is used to refresh the array during normal operation. The REF command can only be executed during the IDLE state, assuring that a refresh operation cannot occur while any cycles are in progress.

Once the refresh operation is completed the machine automatically transitions to the PRECHARGE state before returning to the IDLE state.

Transition to the POWER-DOWN state causes all input circuits to the memory array to become high impedance, effectively cutting off the array from the rest of the device. No refresh operations are performed while the machine is in the POWER-DOWN state. The transition of CKE from high to low along with other input conditions causes the PWRDN command to execute as shown in table 2-1. The machine can remain in the POWER-DOWN state for an indefinite amount of time. A low to high transition on CKE along with other input states causes execution of the PWRDNX command, allowing the machine to return to the IDLE state.

On power-up the mode register is programmed after all internal banks are precharged. Once the machine enters the IDLE state for the first time, execution of the MRS command causes the machine to transition to the MODE-REGISTER-SET state. While in this state the address bus is used to program the register. After the operation is completed the machine automatically returns to the IDLE state. Precharge and burst stop operations cannot be executed from the IDLE state, hence execution of these commands while in the IDLE state has no effect on the state machine.

Read and write cycles each begin with execution of the ACTV command, which causes the machine to transition to the ROW-ACTIVE state. While in the ROW-ACTIVE state the row address is latched and decoded The correct row is. then initialised. While in the ROW-ACTIVE state another ACTV command can be executed, as long as the access is to another bank. Accesses to the same bank are unnecessary and illegal. If the CKE signal switches from high to low at any time during the ROW-ACTIVE state, the machine transitions to the CLOCK-SUSPEND state. While in the CLOCK-SUSPEND state the machine ignores all inputs. The

current operation is not lost but rather is held active and commences following a low to high transition on CKE. This transition causes the machine to return to the ROW-ACTIVE state.

Once the correct row has been initialised, the assertion of CAS# causes the machine to execute one of the following four commands; READ, READA, WRIT, WRITA. Execution of the READ command causes the machine to transition to the READ state. While in the READ state many transitions can occur. If the following cycle is another read to the same bank a READA command can be executed which auto-precharges the bank for the next operation while the current read operation is still in progress, thereby hiding some of the precharge latency. If the read operation is a full-page burst, execution of the BST command causes the machine to terminate the burst and transition to the ROW-ACTIVE state. If the read cycle is allowed to complete and no follow-on cycle is detected, the machine returns to the ROW-ACTIVE state. The READ-SUSPEND state is entered by a high to low transition on the CKE signal and is identical to CLOCK-SUSPEND state. During this state the retrieved data is continually output until the low to high transition on CKE, causing the machine to return to the READ state.

The machine transitions to the WRITE state by execution of the WRIT command. The WRITE state is very similar to the READ state. The WRITE-SUSPEND state performs the same function as CLOCK-SUSPEND. This is also true for the READA-SUSPEND and WRITA-SUSPEND states.

If no other cycles are pending once either a read or write cycle finishes, the machine automatically returns to the ROW-ACTIVE state. However, execution of either a PRE or PALL command from either the READ or WRITE state causes the machine to transition to the PRECHARGE state. The difference between the READ and READA state is that, once the cycle is completed in the READ or WRITE state, an explicit PRE or PALL command must be executed in order for the machine to perform a precharge operation. Whereas in the READA or WRITA state the transition occurs automatically upon completion of the operation.

# 3 SH-4 Bus State Controller SDRAM Overview

The SH-4 series have been designed to provide a direct interface to SDRAM. Once the BSC has been correctly configured, the user simply treats the memory area as RAM, and can perform read/write accesses of 8-, 16-, 32-, 64-bit & 32-byte without being concerned by the physical width of the connected SDRAM. The control logic of the SDRAM interprets transitions on the external pins as commands which are in turn used to transition the state machine. The SH-4 supports a sub-set of these possible SDRAM commands. Furthermore two chip select areas are supported, each providing a 26-bit area, which provides a maximum configuration of 128 Mbytes ( 2 x 2^26) SDRAM total. Where only one area is used, then CS area 3 should be chosen, as it does support RAS Down mode. It is not possible to select area 2 as SDRAM area alone. The supported bus widths for SDRAM are 64- or 32-bit for the 7750 and 32-bit only for the 7751.

The real performance of SDRAM is realised when performing burst accesses. For the SH-4, all burst accesses are of length 32 bytes. This corresponds to 4 words with the bus in 64-bit mode, or 8 words when the bus is set to 32-bit mode.

Looking to the rest of the SH-4 architecture, the 32 byte burst length is mirrored in the cache line length, store queues and the DMAC. The SDRAM commands in the following table indicates in which situations burst accesses are supported.

| Internal access type | Cache mode | SDRAM access | Commands: SH7750, 64 bit data bus | Commands: SH775x/1,32-bit data bus |
|---|---|---|---|---|
| Random read | non cached | single read** | ACTV + ReadA & M | ACTV + Read + ReadA & M |
| Random read* | write through | 32 byte burst | ACTV + ReadA | ACTV + Read + ReadA |
| Random read* | write back | 32 byte burst | ACTV + ReadA | ACTV +Read + ReadA |
| Random write | non cached | single write** | ACTV + ReadA & M | ACTV + Write + WriteA & M |
| Random write | write through | single write** | ACTV + ReadA & M | ACTV + Write + WriteA & M |
| Random write* | write back | 32 byte burst | ACTV + WriteA | ACTV + Write + WriteA |
| DMAC single read | X | single read** | ACTV + ReadA & M | ACTV + Read + ReadA & M |
| DMAC block read | X | 32 byte burst | ACTV + WriteA | ACTV + Write + WriteA |
| DMAC single write | X | single write** | ACTV + WriteA & M | ACTV + Write + WriteA & M |
| DMAC block write | X | 32 byte burst | ACTV + WriteA | ACTV + Write + WriteA |

**Table 3-1 SH-4 SDRAM command**

ACTV: Active command
ReadA: 4 times read with precharge
WriteA: 4 times write with precharge
Read: 4 times read
Write: 4 times write, M
DQM signal masking is used for single read ore write.

*These accesses are always made via the cache using burst mod

**Note that the burst length of the SDRAM is set at startup, and does not change during SH-4 operation. What this means is that **all** SDRAM accesses are made using a burst type cycle. In the case of a single read or write, the wanted data is strobed in using the DQM signals. This means, in the case of a byte write to SDRAM for example, one DQM is active for one of the cycles only, which means no other data is overwritten. The SDRaM burst length setting when connected to the 7751 is half that of the 7750, however a burst access is still 32bytes. See “SDRAM Controller differences between 7750S & 7751” in the next section for details.

In most cases, the cache will be switched on and operates with a write back policy. This ensures maximum CPU/FPU performance, and will mean the SDRAM is accessed using the high efficiency burst access.

In the case of a read cache miss, the first access will be to the missed data, and the rest of the 32byte block will be read in wrap around mode. This increases the efficiency during a cache fill, when the missed data is not aligned to a 32 byte boundary.

32-byte boundary data is also written in wrap around mode.

The basic memory storage technology inside SDRAM is DRAM, and hence memory cells need to be refreshed. There are two refreshing options available, both of which are supported by the SH-4. The selection of the refresh mode is controlled by the RMODE bit in the MCR (mode control register).

Auto refresh is a command which is issued by the BSC. Included in the BSC is a refresh timer, which issue a REF command on a compare match. The user needs to configure this timer such that each row will be refreshed within a certain period (eg. 64ms). In most cases the lost bandwidth from refresh cycles can be ignored (fraction of a percent).

In addition, it is possible to put the SDRAM into Self Refresh mode. The important difference here is that the SH-4 BSC does not need to issue refresh commands as the SDRAM refreshes itself automatically. This is generally used to support system power down, where the SH-4 can disable the clock (standby mode), but the SDRAM contents are not lost. **It should be noted however that when the SDRAMs are set to self refresh, they are not accessible. Special care should be taken that no access to SDRAM is made in cases where self refresh mode is used. (Ie. Exit from standby mode via interrupt, which means the exception handling routine must be in valid memory)**

The performance of SDRAM can further be enhanced using of RAS down mode. Note that this mode is only possible when **area 3 only** is used. SDRAMs are addressed with a row and a column address. Higher speed accesses can be performed within the same row because it is not necessary to change the column address. This is the principle of RAS Down mode. When it is enabled, a row comparator within the BSC automatically checks the current row address to the requested access row address. If there is a miss (different row access), then a standard access consisting of a row + column, is performed. However, if there is a row hit (same row access) th an access without the row cycle is made. In a typical SDRAM configuration this means a saving of 2 CKIO cycles (MCR.RCD=2), which translates to 4 or 6 processor cycles, depending on the clock ratios.

The setting of the RAS Down mode goes with the pipeline access mode. Pipelined access is performed between an access by the CPU and an access by the DMAC, or in the case of consecutive accesses by the DMAC, to provide faster access to synchronous DRAM. As synchronous DRAM is internally divided into two or four banks, after READ or WRIT command is issued for one bank it is possible to issue a PRE, ACTV, or other command during the CAS latency cycle or data latch cycle, or during the data write cycle, and so shorten the access cycle.

# 4 SDRAM Controller differences between SH7750S & -7751

## 4.1 Overview

The SH7750S & SH7751 have very similar BSC SDRAM control features, however there is one significant difference between the two devices for the basic SDRAM bus cycle. This is that the burst length of SDRAM. When you connect SDRAM to the 7751 with 32-bit bus, your should set a burst length of 4, as compared with 8 for the 7750. This means 2 reads or 2 writes are required to complete a 32 byte access (compared to 1 for the 7750S), for example to fill or write back a cache line.

| | 7750S | 7751 |
|---|---|---|
| SDRAM Burst length 32bit width | 8 | 4 |
| SDRAM Burst length 64 bit width | 4 | - |
| Number of READ/WRIT commands for a burst access | 1 | 2 |
| Burst access size (=cache line size) | 32byte | 32byte |

**Table 4-1: Summary of Differences between SH-4 Device SDRAM Controllers**

It should be noted that the need to issue two reads or two writes to complete a 32 byte access, does not impose any bus overhead compared with a burst read/write issued from a single command, as with the 7750S. The reason for this is that the SDRAM accesses are pipelined, so that the total number of bus cycles does not change. The below diagram shows this. Note that in particular the two READ commands, signified by CAS = low.

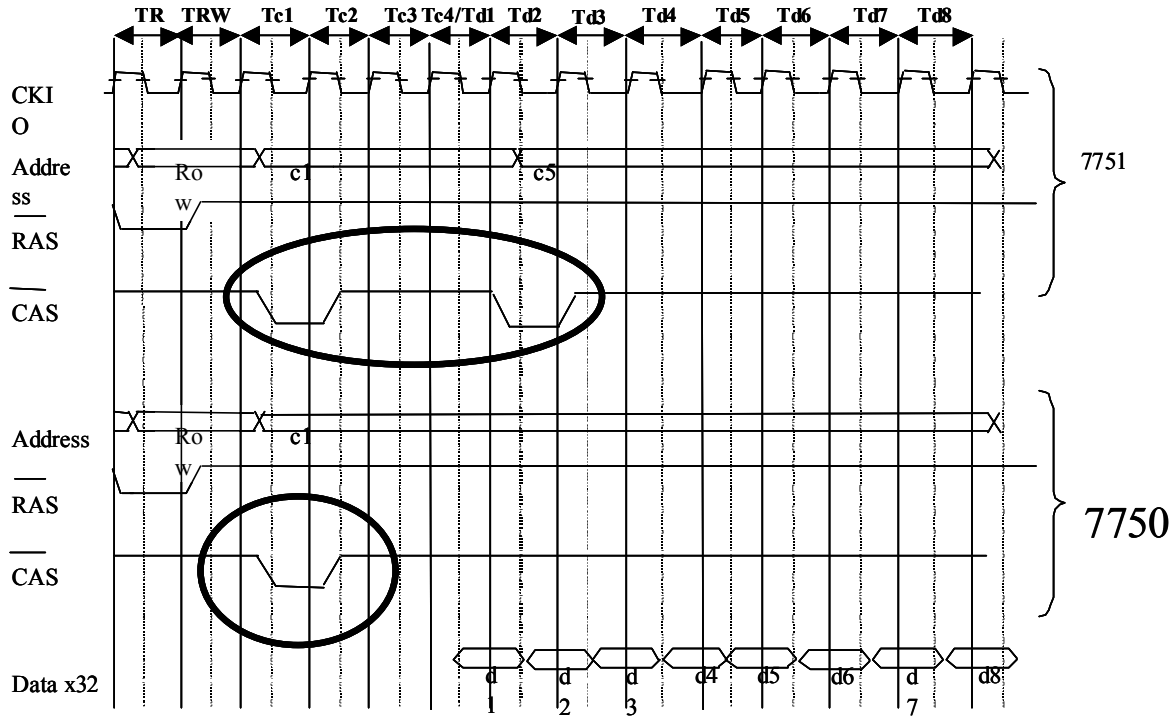


**Figure 4-1 7750S & 7751 SDRAM 32byte Burst Read from 32-bit Wide Memory**

SH-4 Bus State Controller Registers relating to SDRAM

When SDRAM is used with the SH-4, several BSC registers must be configured before the normal SDRAM operations can be performed.

Below there is a summary table showing which registers need to be configured in order to access SDRAM. Details of each register can be found in the relevant SH-4 Hardware user manual.

| BSC Register | Width | Relevance to SDRAM control |
|---|---|---|
| BCR1 | 32 | Selects SDRAM memory for area 3, area 3 & 2 or none |
| BCR2 | 16 | Used to select databus width of SDRAM and other areas |
| WCR1 | 32 | Idle cycles to be inserted following access to each CS area, |
| WCR2 | 32 | CAS latency setting (1-5 cycles). |
| WCR3 | 32 | No effect on SDRAM area |
| MCR | 32 | Specifies RAS & CAS timing, address multiplexing and refresh control, critical SDRAM configuration |
| PCR | 16 | No effect on SDRAM area |
| RTCSR* | 16 | Controls refreshing settings |
| RFCR* | 16 | Counts the number of refreshes issued |
| RTCNT* | 16 | Refresh timer counter |
| RTCOR* | 16 | Refresh compare match register |
| SDMR2 | 8 | Virtual write only register, controls CAS latency & burst length. Physcially exists inside the SDRAM |
| SDMR3 | 8 | Virtual write only register, controls CAS latency & burst length. Physcially exists inside the SDRAM |

**Table 4-2: SH-4 Bus State Controller Registers related to SDRAM**

*Note: The refresh control registers are all password protected to reduce the chance of accidental writes. A special value must be included in the upper bits to allow a write value to be latched

## 4.2 Bus State Controller Register

### 4.2.1 Bus Control Register 1 (BCR1):

Bits 4..2 – Controls the memory type of CS area 2 & 3. Note that the BSC will allow SDRAM in area 2&3, area 3 alone or neither. It is not possible to select SDRAM memory type just for area 2. Hence always use area 3 if only one area of SDRAM is used.

### 4.2.2 Bus Control Register 2 (BCR2):

Bit 0 – Controls the use of D51 to D32. These bits function as a port if set to 1. In this mode 64bit bus width is illegal

Note: the width settings defined in area 2 & 3 are ignored when SDRAM is enabled. The MCR register setting controls the width for the SDRAM areas.

### 4.2.3 Wait Control Register 1 (WCR1):

Bits 14..12 – Controls the number of idle cycles inserted following an access to CS area 3. The hold time of SDRAM by definition is very short (less than 1 CKIO cycle). In most cases, this can be set to 0.

Bits 10..8 – Controls the number of idle cycles inserted following an access to CS area 2. The hold time of SDRAM by definition is very short (less than 1 CKIO cycle) so in most cases, it can be set to 0.

### 4.2.4 Wait Control Register 2 (WCR2):

Bits 15..13 – Sets the CAS latency of CS area 3, when SDRAM is enabled. CAS latency is effectively the number of clock cycles after a read command, that the first data can be strobed in

by the SH-4. This can be set from 1-5 cycles, or 2/3 cycles when RAS Down mode is enabled. Other settings are not guaranteed. Note that the CAS latency setting must be the same value as written to the SDMR (internal SDRAM register), to ensure correct sampling of read data.

Bits 11..9 – Sets the CAS latency of the CS area 2, when SDRAM is enabled. Note that when areas 2 & 3 are used as SDRAM, RAS Down mode is not supported.

### 4.2.5 Wait Control Register 3 (WCR3):

WCR3 has no effect upon the operation of SDRAM.

### 4.2.6 Individual Memory Control Register (MCR):

MCR is a 32-bit read/write register that can be used to specify RAS and CAS timing, address multiplexing and refresh control for SDRAM in Area 2 and 3.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RASD | MRSET | TRC2 | TRC1 | TRC0 | - | - | - | TCAS | - | TPC2 | TPC1 | TPC0 | - | RCD1 | RCD0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | TRWL2 | TRWL1 | TRWL0 | TRAS2 | TRAS1 | TRAS0 | BE | SZ1 | SZ0 | AMEXT | AMX2 | AMX1 | AMX0 | RFSH | RMODE | EDO |

**Table 4-3 WCR3 Register**

Bit 31 – **(RASD)** Controls **RAS Down mode** operation. A normal SDRAM access consists of a row address, followed by a column address, which are multiplexed on the same address bus in turn. These two cycles specify a complete address and are necessary for most random accesses. In the case of an access to the same row as before, and internal comparator can force the BSC to inhibit the ROW address, thereby speeding up the access by tRCD. This effectively means the SDRAM state machine is normally in the row active state, rather than the idle state. This mode can only be used when area 3 is the only SDRAM area.

RAS Down mode is recommended for maximum performance.

Also when RASD bit is set to 1 in MCR, **pipelined access** is performed between an access by the CPU and an access by the DMAC, or in the case of consecutive accesses by the DMAC, to provide faster access to synchronous DRAM. As synchronous DRAM is internally divided into four banks, after a READ or WRIT command is issued for one bank it is possible to issue a PRE, ACTV, or other command during the CAS latency cycle or data latch cycle, or during the data write cycle, and so shorten the access cycle.

Bit 30 – **(MRS)** Mode Register Set. Two commands can be issued to the SDRAMs manually by writing to a special area (H'FF90XXXX). The command which is sent upon the write, depends on the setting of the MRS bit. If it is set to 0, then an 'all bank precharge' occurs. If it is set to 1, then a 'mode register set' command is sent, along with the set value which is encoded into the lower address bits.

Bit 29..27 – **(TRC2–TRC0)**. These bits set the number of wait cycles following a refresh cycle (auto or self) to allow for the refresh to complete. Settings are 0,3,6,9,12.. etc.

Bits 21..19 – **(TPC2–TPC0)**. RAS Precharge Period. These bits specify the minimum number of cycles until the next bank active command is output after precharging. Valid settings are normally 1-8, or 2,3 when RAS Down mode is enabled. (See bit 31)

Bits 17..16 – **(RCD1, RCD0).** RAS-CAS delay. This sets the number of cycles after a ROW address has been issued that a column address can be issued (Ie. Read/Write). Settings of 2-4 cycles are valid, or 2,3 cycles in the case of RAS Down mode (bit 31). By adding the RAS-CAS delay tRCD to CAS latency, the random access read time can be calculated.

Bits 15..13 – **(TRWL2–TRWL0)**. These bits set the SDRAM write to precharge delay time. Basically the write operation has to complete internally before a precharge can be issued. This setting forces the BSC to wait following a write command. Valid settings are 1-5 cycles, or 1,2 cycles in RAS Down mode (see bit 31).

Bits 12..10 – **(TRAS2–TRAS0)**. Auto refresh delay. These bits define how long the BSC will wait following an refresh command. Valid setting are 4-11, although the actual delay used is added to the TRC value, which is set by bits 29..27.

Bits 8..7 – **(SZ1, SZ0)**. Here, the width of the SDRAM interface is defined. These bits take priority over the setting in BCR2. For the 7750, a setting of 64 or 32bits is valid, and for the 7751, 32-bits is the only supported SDRAM area width.

Bits 6..3 – **(AMXEXT, AMX2–AMX0)**. The address multiplexing is controlled by these bits. During the RAS cycle, the upper address bits are shifted down by the number of column bits, so that they appear at the portion of the address bus connected to the SDRAM. During the CAS cycle, no multiplexing occurs. The setting of these 4 bits depends on the organisation of the SDRAM connected. The SH-4 Hardware User Manual provides SDRAM multiplexing tables in the appendix, or you can refer to the Address Multiplex Guide in this application note.

Bit 2 – **(RFSH)**. Refresh control. This bit switches on or off the refreshing of the SDRAMs.

Bit 1 – **(RMODE)**. This bit controls the refreshing mode. When this bit is set to 1, self refresh mode is entered. It basically issues a SELF command, which also drives the CKE (clock enable signal) to the SDRAMs. During this state, the SDRAMs do not exist in the memory map, so it is important to execute this code from a different area (eg. FLASH).

### 4.2.7 PCMCIA Control Register (PCR):

PCR has no effect upon the operation of SDRAM.

### 4.2.8 Refresh Timer Control/Status Register (RTSCR):

Note: The password (upper 8 bits of 16-bit word) to write to this register is – B'10100101 (H'A5)

Bit 7 – **(CMF)** Indicates a RTCNT, RTCOR compare match. Does not need to be initialised during SDRAM configuration.

Bit 6 – **(CMIE)** Compare match interrupt enable. Should normally be set to 0.

Bits 5..3 – **(CKS2 – CKS0)** Clock prescaler configuration bits. These bits set the division ration of the refresh counter clock, which is derived from CKIO.

Bit 2 – **(OVF)** Overflow flag which can be configured to be set following 512 or 1024 refreshes.

Bit 1 – **(OVIE)** Interrupt control bit. Enables or disables the interrupt after 512 or 1024 refreshes as set by bit below. Normally this bit (and hence interrupts) is disabled, as refreshing is taken care of by the refresh timer hardware.

Bit 0 – **(LMTS)** Specifies whether the OVF flag is set following 512 or 1024 refreshes

### 4.2.9 Refresh Count Register (RFCR):

Note: The password (upper 6 bits of 16-bit word) to access this register is – B'101001 (H'A4)

Bits 9..0 – **(RFCR 9..0)** This register keeps a count of number of refreshes and clears itself when = 512 or 1024 as set by RTCSR.LMTS. Generally this register is not of interest, apart from during initialisation, when a pre-set number of refreshes need to be waited for.

### 4.2.10 Refresh Timer Counter Register (RTCNT):

Note: The password (upper 8 bits of 16-bit word) to access this register is – B'10100101 (H'A5)

Bits 7..0 – **(RTCNT 7..0)** This register is the actual refresh timer counter. It is 8 bits wide. When RTCNT=RTCOR, a refresh cycle request is made and RFCR is incremented.

### 4.2.11 Refresh Time Constant Register (RTCOR):

Note: The password (upper 8 bits of 16-bit word) to access this register is – B'10100101 (H'A5)

Bits 7..0 – **(RTCOR 7..0)** This register is the 8 bit compare match value register. When RTCNT=RTCOR, a refresh cycle request is made and RFCR is incremented

### 4.2.12 Synchronous DRAM Mode Register 2 & 3(SDMR2, -3):

The SDRAM mode register is a configuration register that physically resides within each SDRAM. It controls CAS latency, burst length and wrap type of the SDRAM. CAS latency must be set to the same value in the MCR, and specifies how many clocks after a READ, data is output. Burst length is the number of data frames that will be read/written in each access. Table 4-4 shows the burst length required by the SH-4. Depending on which device and what bus width is used, a different value will need to be set.

| Device/Configuration | Burst length |
|---|---|
| SH7750S / 32bit bus | 8 |
| SH7750S / 64bit bus | 4 |
| SH7751/32bit bus | 4 |

**Table 4-4: Required Burst Length Setting for SDMR**

Writing to mode register uses the address bus rather than the data bus. The 'Mode Register Set' command is issued to the SDRAM manually by writing to a special area (H'FF90XXXX), when the MRS bit in the MCR is set to 1. The set value for the SDMR is encoded into the lower address bits as follows:

If the value to be set is X and the SDMR address is Y, the value X is written in the SDRAM mode register by writing at address X + Y. Since for 32-bit bus, A0 of SDRAM is connected to A2 of the processor and A1 of SDRAM is connected to A3 of the processor, X value must be shifted two bits right before being added to Y. For the SH-4, the Y value for SDRAM in Area 3 is H'FF940000, and in Area 2 is H'FF900000.

Similar procedure is possible with 64-bit bus. However it is more simple to reference the SH-4 Hardware Manual. It contains already all possible combination which could be set for SDRMR, see Table 4-5.

| Bus Width | CAS Latency | Area 2 | Area 3 |
|---|---|---|---|
| 32 | 1 | H'FF90004C | H'FF94004C |
| | 2 | H'FF90008C | H'FF94008C |
| | 3 | H'FF9000CC | H'FF9400CC |
| 64 | 1 | H'FF900090 | H'FF940090 |
| | 2 | H'FF900110 | H'FF940110 |
| | 3 | H'FF900190 | H'FF940190 |

**Table 4-5 possible SDMR settings**

# 5 Connecting the SH-4 directly to SDRAM

## 5.1 Physical connection

Like all memories, SDRAMs require some method of specifying an address when reading or writing data and control lines to co-ordinate the operation. The table below provides a functional summary of the groups of signals present on a typical SDRAM. The 'Pin arrangement' section of the SH-4 HW manual also provides a useful guide to which pins are used for the SDRAM (and other memory type) interface.

| SDRAM signal | SH4 Signal | Purpose | Notes |
|---|---|---|---|
| CLK | CKIO | Provides clock to the SDRAM state machine, rising edge triggered | Critical signal - Clock routing important |
| CKE | CKE | Clock enable - freezes operation of SDRAM | used during power down, PLL stepping |
| CS | CSx | Chip Select | |
| Address | Address | Multiplexed address input, also forms part of command word during ACTV | Multiplex settings discussed below |
| BA 0..N | Address | Bank select, effectively an upper address specification | Normally 4 bank |
| RAS | RAS | Row address strobe, latches row address | |
| CAS | RD | Column address strobe, latches column address | also issues read/write |
| WE | RD/WR | Selects either read or write from CAS during cycle | |
| DQ 0..N | D 0..M | Data input / output | |
| DQM 0..N | DQM | Data qualifier masks. Specifies which bytes should be written during write | |
| VDD/Q | | Power supply | |
| VSS/Q | | Power supply ground | |

**Table 5-1: Interface, SH-4 and SDRAM**

Most of the signals above are analogous to a SRAM. DQ (or data) lines are the data input/output port, DQM are similar to the HWR/LWR or HBS/LBS data masks, and the WE is a read/write strobe. CS is a chip select and the CLK/CKE lines are clock control for the SDRAM, as it is essentially a synchronous state machine. The address lines, coupled with the bank select and RAS/CAS lines are slightly more complex. They provide the full address over two phases. This is described in the next section.

### 5.1.1 Address Multiplexing

To keep the pin count low on SDRAMs, the address is multiplexed. Eg. in the case of a 128-Mbit x16 SRAM device, a 23-bit address would be required, however only 12 address bits and 2 bank selects are provided by the SDRAM.

To provide the complete address to the SDRAM, a RAS and CAS phase are required. The table below shows how this is split for a typical 128-Mbit device.

| 64bit, 4 x 128Mbit (x16) devi | 12bit row | 9bit col | | AMX3 |
|---|---|---|---|---|
| SH7750 terminal | RAS cycle | CAS cycle | SDRAM terminal | Function |
| A18 | X | X | | X |
| A17 | X | X | | X |
| A16 | A25 | A25 | BA1 | BANK select address 1 |
| A15 | A24 | A24 | BA0 | BANK select address 0 |
| A14 | A23 | X | A11 | Address |
| A13 | A22 | L/H | A10 | Address / precharge set |
| A12 | A21 | X | A9 | Address |
| A11 | A20 | A11 | A8 | Address |
| A10 | A19 | A10 | A7 | Address |
| A9 | A18 | A9 | A6 | Address |
| A8 | A17 | A8 | A5 | Address |
| A7 | A16 | A7 | A4 | Address |
| A6 | A15 | A6 | A3 | Address |
| A5 | A14 | A5 | A2 | Address |
| A4 | A13 | A4 | A1 | Address |
| A3 | A12 | A3 | A0 | Address |
| A2 | X | X | Not used | X |
| A1 | X | X | Not used | X |
| A0 | X | X | Not used | X |

**Table 5-2: Table showing address multiplexing during RAS & CAS cycle**

The first column represents the physical pins of the SH-4, in this case a 7750 operating with a 64-bit wide bus (Hence A2..A0 on the SH-4 are unused). The second and third columns shows which signals actually get presented on the address bus during the RAS and CAS cycle. The so called mapping here is 12/9. The fourth column shows the SDRAM address pins. The important features are as follows:

**RAS Cycle**

A25..A12 – These are the upper 12 bits which specify the row address

BA1..BA0 – The bank select bits decide which of the four banks is selected

**CAS Cycle**

A11..A3 – These are the lower 9 bits of the address, which select the column address

A10 – A10 becomes the 'Precharge Select bit'. This bit does not specify any address information. Instead it makes up part of the command word. The level of the precharge select bit defines whether an auto precharge, or standard read/write is performed. Auto precharge accesses are always used on the SH-4 unless RAS Down mode is enabled.

BA1..BA0 – The bank bits decide which of the banks is selected

Note: The bank select bits do not change from the RAS to the CAS cycle. This effectively means these bits are not multiplexed. Note that the same address line is output during the RAS and CAS cycle.

Summing the address lines together over the RAS and CAS cycle for the 128-Mbit x16 device mentioned, you get:

| | | |
|---|---|---|
| Total address bits | = 12 (Row) + 9 (Column) + 2 (Bank select) | = 23 bits |
| Total capacity | = 2^23 = 8M words (device is organised x16) | = 16 Mbytes |

The timing diagram below shows the address, bank select and precharge select bits during a SH7750, 32-bit wide SDRAM burst read. Note that the Bank bits remain static through the RAS & CAS phases, and the precharge bit represents an address when RAS is asserted, and part of the command word (precharge select) during the CAS cycle.

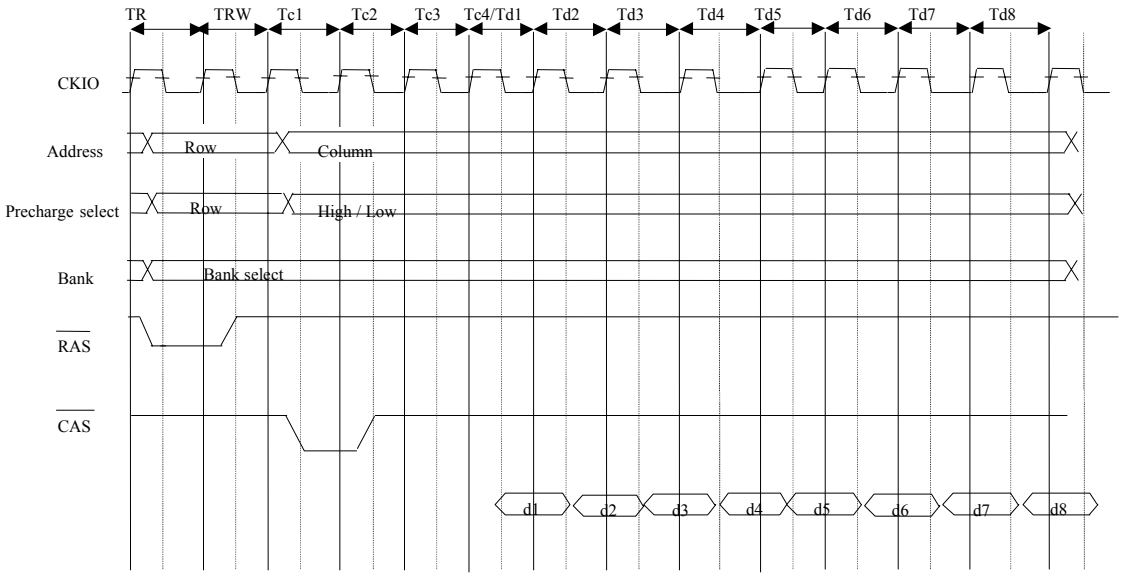


**Figure 5-1: Timing diagram, address and precharge bits**

Clearly providing the correct address multiplexing setting is important. The address multiplexing is controlled by setting the AMX bits in the MCR (memory control register). On the SH-4, there are 4 AMX bits, AMX 3..0 and AMXEXT. The SH-4 Hardware user manual provides SDRAM multiplexing tables in the appendix.

For each design it is recommended that the address multiplexing is double checked, so that all of the signals are correctly output during both RAS and CAS cycles.

## 5.2 Timing Check

Performing a timing analysis between a SDRAM and the BSC initially seems very difficult, especially following a look at timing diagrams. There are many parameters to analyse and meet, however the task is not so much difficult as lengthy.

The method of timing analysis used within this application note is to take each of the SDRAM parameters in turn, and attempt to meet them by equating them to the SH-4 BSC parameters. Below is a description of each of the parameters specified by a typical SDRAM. As there are many timing diagrams required to illustrate these parameters, it is recommended that a copy of your SDRAM datasheet is consulted whilst this section of the application note is read.

At the end of section 5.2 there is a timing exel sheet which summarises all related timings, equations and results. It gives you a easy to handle timing sheet for a quick and comprehensive timing check.

The type of timing analysis to be made depends on the parameter being checked. It can be useful to differentiate between the four types of timing parameters specified by the SDRAM, which must be met by the SH-4 BSC. These are as follows:

### 5.2.1 Setup and Hold times

Because the SDRAM is a state machine, state transition commands must be provided by the SH-4 BSC. Each state transition occurs on the **rising edge** of CKIO, and hence the command word is sampled on the positive edge of CKIO. Any signal that is sampled has an associated setup and hold time, so these parameters must be met. In addition, any read or write will involve sampling of data, either by the SH-4 BSC or the SDRAM. This also happens on the positive edge.

### 5.2.2 Wait states

The wait state control within the SH-4 BSC essentially sets the state transition timing. This means all wait state controls specify an integer number of clock cycles. In all cases the clock used for timing calculations, is the bus clock or CKIO, and hence the values are multiples of the CKIO period. Some values are given in cycles, for a particular frequency (eg. 66/100 MHz. In the case when a specification is not available within the SDRAM datasheet for your chosen clock frequency, the next highest frequency condition specification should be chosen. Eg. Use 100 MHz figures if you are designing with an 83 MHz bus, when the SDRAM does not specify a 83 MHz test condition. Most of the SDRAM datasheet parameters fall into this category. In general the parameters to be met here are minimum values, and hence the wait states must be increased until the parameter is met or exceeded.

### 5.2.3 Auto Refresh interval timing

When the SDRAM is active, the task of refreshing the DRAM cells is controlled by the refresh timer within the BSC. When the internal timer makes a compare match, the SH-4-BSC will issue the relevant commands to perform an auto refresh. How often this needs to be done needs to be calculated, and the appropriate refresh controller registers must be configured.

### 5.2.4 Clock Quality

The clock speed for SDRAM is very fast. This means a strong clock drive is required so that the clock provided is close to 50% duty, and has good rise and fall times. The parameters of interest here are rise & fall times, worst case high and low times and voltage threshold levels. It is also very important to consider the rest of the system here, as the CKIO frequency and quality is often limited by other devices connected to the SH-4 local bus, and the PCB layout.

Probably the most important, and usually first, decision to be made is the choice of **bus clock frequency**. In the case of a 200 MHz (feasible with 7750S and package BGA), likely choices are 100 Mhz (CKIO = CPU/2) or 66 MHz (CKIO=CPU/3). For a 167 MHz SH-4 such as the SH7751 likely choices are 83 MHz (CKIO = CPU/2) or 55 MHz (CKIO=CPU/3). Choice of system bus speed is extremely important in all systems. Performance requirements generally mean high speed, and many factors require lower speed, such as EMC/EMI, timing problems, power dissipation and other limiting factors such as peripheral ICs. Note that the SH-4 supports dynamic clock scaling so low power can be achieved by modifying the CPU/BUS/PERIPHERAL clock when the system is in idle state.

Generally higher clock frequencies mean higher performance, and this is especially the case where pure bandwidth is concerned. However in the case where several random accesses may occur, or when a read cache miss occurs, the time to first access is often the key factor. Note that sometimes it is possible (depending on SDRAM settings) to reduce the CAS latency from 3

cycles to 2 cycles, when using a lower bus frequency. This can compensate for the lower bus frequency and provide a reasonable random access read time.

### 5.2.5 SDRAM Setup and Hold parameters

Note: In all cases $t_{CYC}$ = CKIO period.

Eg. 100 MHz = 10ns, 83 MHz = 12ns, 66 MHz = 15ns, 56 MHz = 18ns and 50 MHz = 20ns

For better imagniation you can find most timing values also indicated in the timing diagram Figure 5-2.

#### 5.2.5.1 Tac

Tac is the access time of read data, from the rising edge of the clock. This means it is the **maximum** delay to valid data from a rising edge. Note that this parameter is not related to CAS latency. This parameter can be used to calculate **setup time for read data** for the SH-4. Because the data is sampled on the following rising edge (1 CKIO cycle later), the SH-4 equation, and hence condition to meet is:

SDRAM : SH-4 Tac =< (tCYC-tRDS)

#### 5.2.5.2 Toh

Toh is the read data output hold time. It specifies the **minimum** time that the SDRAM will hold valid data following the rising edge of a read cycle. This equates directly to the SH-4 BSC required **read data hold** time. Hence the following condition must be satisfied.

SDRAM : SH-4 Toh => tRDH

#### 5.2.5.3 Tsi

Tsi is **the input setup times** of the SDRAM control signals (such as CS, ADDR, RAS, CAS, BS & DQM), the **input setup time for write data** and the input setup time for CKE, when the clock is suspended. These setup times have got different naming in the SDRAM data sheet however the same value. Hence it will be references as Tsi. The parameter will depend on the **CKIO cycle time** and **maximum output delay time** from the SH-4 BSC. The equations that must be satisfied are as follows:

SDRAM : SH-4 $t_{ADRESS\_SETUP\_SDRAM} =< (t_{CYC}-t_{WED1})$

$t_{CHIP\_SELECT\_SETUP\_SDRAM} < (t_{CYC}-t_{RASD})$

etc.

Note: Since in the SH-4 datasheet all output delays are identical, and in the SDRAM datasheet all input setup times are identical, only one of these equations needs to be calculated to be sure of meeting all SDRAM setup parameters.

#### 5.2.5.4 Thi

tXXH are **the input hold times** of the SDRAM control signals (such as CS, ADDR, RAS, CAS, BS & DQM), the **input hold time for write data** and the input setup time for CKE, when the clock is suspended. This parameter will depend on the **CKIO cycle time** and **minimum output hold time** from the SH-4 BSC. The equations that must be satisfied are as follows:

SDRAM : SH-4

$t_{DATA\_HOLD\_SDRAM} =< t_{WDD}$

$t_{CHIP\_SELECT\_HOLD\_SDRAM} =< t_{CSD}$

etc.

Note: Since in the SH-4 datasheet all output delays are identical, and in the SDRAM datasheet all input hold times are identical, only one of these equations needs to be calculated to be sure of meeting all SDRAM hold time parameters.

### 5.2.5.5 tLZ

tLZ is the **minimum delay to the data bus being driven during a read**. This parameter will always be met unless there are other devices in the system which are likely to drive the databus long beyond the end of their bus cycle. It is generally recommended that, when very slow devices are connected to the system, idle cycles are configured, or buffers inserted to avoid holding the databus for too long. In most systems there will be a **worst case data bus hold time** following a read, and as long as this is less than a bus cycle, tLZ will always be met. This is especially the case as the fastest the data bus will be driven by the SDRAM will occur during a RAS Down read, which still has to wait for the CAS latency time Tcas (2 or3) before it drives the bus. The actual equation to meet is:

SDRAM : SH-4 $t_{LZ}$ =< ((Tcas –1) *$t_{CYC}$ )- system worst case databus hold time)

Tcas settings are made in register WCR2.

### 5.2.5.6 tHZ

tHZ is the **maximum data bus hold time** of the SDRAM following a read. This is parameter is also important to consider when designing the system to avoid data bus collisions. The worst case here is an SDRAM read followed by a write from the SH-4. Because there is no minimum delay time specified for output data during a write, if tHZ is greater than zero then it is necessary to insert some idle cycles (normally 1). Hence idle cycles must be taken into consideration in this equation:-

SDRAM : SH-4 $t_{HZ}$ =< ((idle cycle setting * $t_{CYC}$)

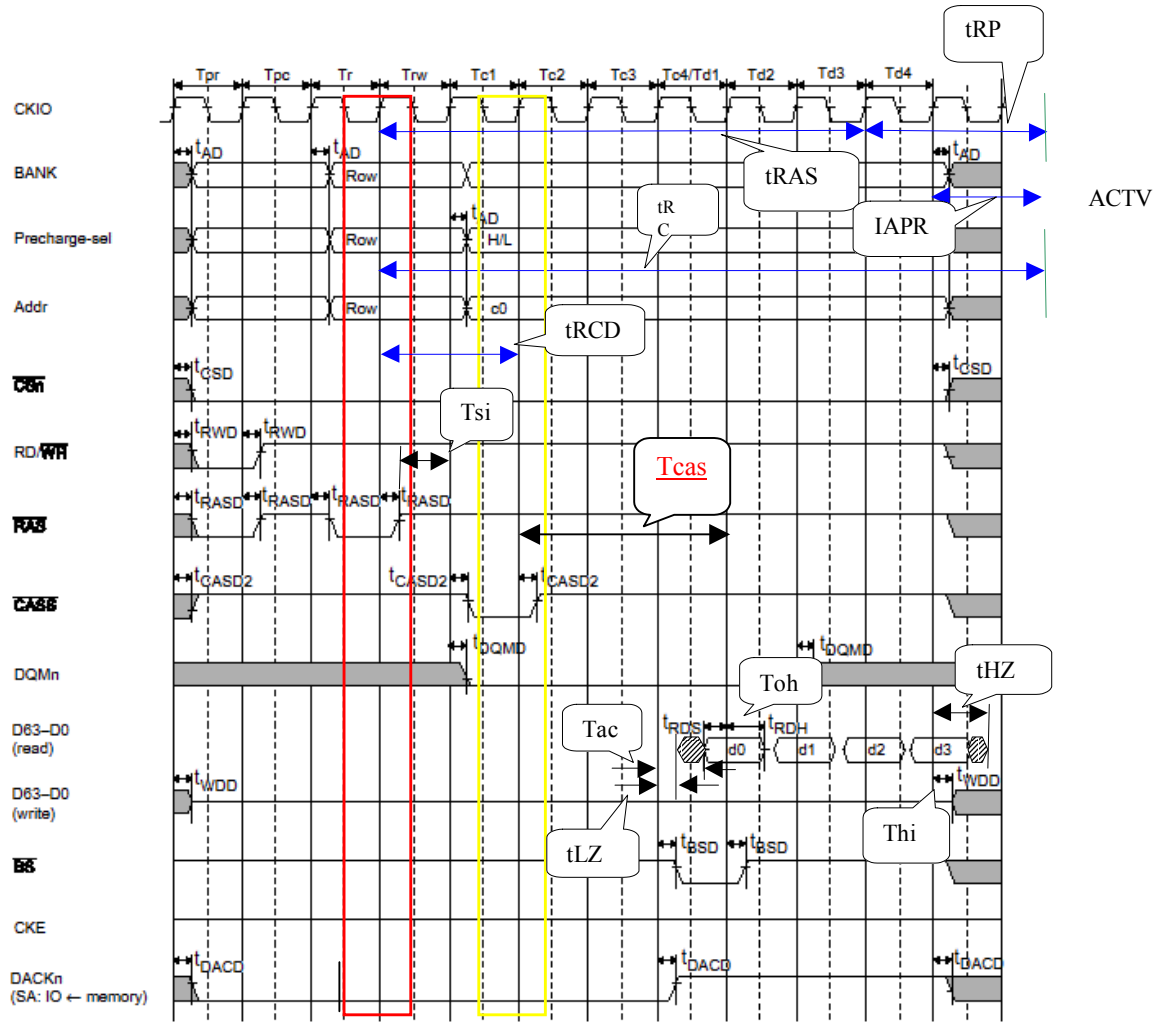


**Figure 5-2: Combined SDRAM and SH-4 Timings. SH-4-Timings: SH-4 HW-Manual, Synchronous DRAM Normal Read Bus Cycle: PRE + ACT + READCommands, Burst (RCD[1:0] = 01, TPC[2:0] = 011, RCD = 1, CAS Latency = 3)**

### 5.2.6 SDRAM Wait State parameters

All of the wait state parameters are controlled by adjusting the BSC registers. The goal is to minimise all register values to minimum wait times, whilst still meeting every timing parameter. As before,

$t_{CYC}$ = CKIO period.

#### 5.2.6.1 Trrd

Trrd is the minimum delay between RAS and the next RAS cycle. Ie. The **minimum time period between two ACTV commands of different banks**. A typical value for this parameter is 20ns. Since the SH-4 will always perform a read or write following an ACTV, before issuing the next ACTV, even with CKIO=100 MHz this parameter will always be met.

#### 5.2.6.2 Trcd

Trcd is the delay between the RAS and CAS cycle, or **minimum delay between ACTV and READ/WRIT.** The setting of MCR.RCD controls this value. Hence the equation to meet is:-

SDRAM : SH-4 Trcd =< ($t_{CYC}$ * MCR.RCD)

#### 5.2.6.3 Trp

This is the minimum row precharge time. It specifies the **minimum delay between a PRE command and the next ACTV.** This value will need to be considered for the following cases:-

End of auto-precharge read/write, where an implicit precharge is executed, and in the case of the PRE command being output. (Eg. In the case of a row change during RAS Down access). In both cases the setting of MCR.TPC is key.

Therefore there is one equation correct for both situations:-

SDRAM : SH-4 Trp =< ($t_{CYC}$ * MCR.TPC)

#### 5.2.6.4 Tras (1)

Tras is one of the few parameters that specifies a minimum and a maximum time. In the case of the minimum value, it gives the **minimum time between ACTV and the next PRE.** This value will be governed by how quickly a full read or write (with ROW and COLUMN address) can be executed following the previous one. Since the time between consecutive full read/writes is:

RAS-CAS delay (min 2) + CAS latency (min 2) + Burst length (min 4) = 8 CKIO cycles, and typical Tras requirement is 50ns, even at CKIO = 100 MHz, this parameter is always met.

In the case of the maximum value of Tras, this defines **the maximum time a row can be active without a precharge.** When RAS Down mode is off, this parameter can never be violated because a precharge will automatically be issued after each read/write, and hence the row will be deselected. When RAS Down mode is on, it is possible that the same row may be active for a

long time (e.g. Idle loop in software), and hence you must take care that a command is issued within this period. The way this is normally done is by adjusting the refresh period to Tras. See the 'Refresh settings' section for details.

#### 5.2.6.5 Tdpl

This is the last data write to precharge. This specifies the **minimum delay from when the last data being written is latched, to the next precharge command**, which will happen when RAS Down mode in enabled. The reason for this recovery time is that following latching of the write data, it takes the SDRAM some time before the data is finally written to the internal DRAM cell.

This setting is controlled by the three MCR.TRWL bits. The equation to be satisfied are as follows:-

SDRAM : SH-4 Tdpl =< ($t_{CYC}$ * MCR.TRWL)

#### 5.2.6.6 Tdal

This is the last data write recovery time, until next ACTV. This specifies the **minimum delay from when the last data being written is latched, to the next ACTV command**. The worst case condition for this is after a write (RAS Down off, or RAS Down on with different row access following). Since between the last data being written, and the next ACTV command being issued, a precharge is required this value should be met by the sum of two values.

In addition to the MCR.TRWL bits as above, the settings of the MCR.TPC bits must be considered. The equation to be satisfied are as follows:-

SDRAM : SH-4 Tdal =< ($t_{CYC}$ * (MCR.TRWL + MCR.TPC))

#### 5.2.6.7 Trc

Trc is the minimum row cycle time. This specifies both **the minimum time between a REF/ACTV command** and **the next REF/ACTV command.** Because the refresh interval will typically be more than hundreds of microseconds and Trc is typically 70ns, REF-REF will always be met. The fastest two ACTV command (or ACTV to REF) can be issued will be in the case of two consecutive auto precharge single reads. Since the time for this is:-

RAS-CAS delay (min 2) + CAS latency (min 2) + Burst length (min 4) + Precharge (typ 2) = 10 CKIO cycles, and Trc is typically 70ns, and so this is fine also.

The only issue here is the time from REF to ACTV. Since an ACTV can be output at the beginning of a normal read/write, there must be sufficient delay following the REF command. The setting for this time is defined by MCR.TRAS. The following equation must then be satisfied:

SDRAM : SH-4 Trc =< ($t_{CYC}$ * (MCR.TRAS)

#### 5.2.6.8 Troh

This value is the delay between a precharge command and high impedance of the data bus. This means it is the **minimum time following a precharge that data should be written**. Since following a precharge, the minimum delay before data can be written by the SH-4 is:-

Tpc (min 1) + RAS-CAS delay (min 2) = 3 cycles,

and Troh is minimum 3, this value is always met.

#### 5.2.6.9 Tccd

Tccd is the minimum delay between column to column commands. Effectively, this specifies **the minimum delay between a read/write and the next read/write**. As this parameter is normally specified at 1 cycle, and the fastest the SH-4 can issue to column commands, is the case of the 7751 burst access. Here a read/write will be followed by another read/write 4 cycles later. Hence meeting Tccd always happens.

#### 5.2.6.10 Tdwd/Tdqm

This is the write command to data in latency, and the DQM to data in latency. Since a write consists of both the write command, and DQM to specify which bytes within the word must be written, these parameters can be grouped together. So together, these define the **delay between a write operation and write data being latched.** The SH-4 always assumes this delay to be 0.

Ie. Data is latched upon the write command. This is normal for all normal SDRAMs so this parameter is met.

#### 5.2.6.11 Tdqz

Tccz is the **minimum delay between valid DQM and reading valid data**. Since during a read, the SH-4 will assert all DQMs when the read command is issued, this parameter will always be met as it is less than or equal to the CAS latency.

### 5.2.6.12 Tsrx/$I_{SREX}$

Tsrx is the self refresh exit time, and $I_{SREX}$ is the self refresh exit to command input delay. Tsrx is not really of interest because, $I_{SREX}$ is much greater (typically 7 vs. 1 cycle) and hence meeting $I_{SREX}$ will satisfy both requirements. So basically, the important parameter to meet is **the minimum number of cycles between exiting self refresh, and the first command.** The SH-4 hardware manual provides a single diagram showing entry and exit from self refresh mode, and it can be seen that a number of wait cycles as specified by MCR.TRC are inserted. Hence the equation to meet is:-

SDRAM : SH-4 $I_{SREX} =< (MCR.TRC)$

### 5.2.6.13 Tcke

This is the time it takes the SDRAM to disable the clock following de-assertion of CKE. Since the SH-4 does not issue any commands following de-assertion of CKE (entry to self refresh mode), this parameter is always met.

### 5.2.6.14 Tmrd

Tmrd is the **minimum delay between a MRS and ACTV command.** Following an SDRAM mode register set cycle, the number of wait states before the next command can be issued is fixed. The hardware user manual shows the same timing for a PRE + PALL and PRE + MRS. The wait after the precharge select bank is always 3 cycles and the wait after the PALL/MRS is always 4 cycles. Hence the equation to satisfy Tmrd is:

SDRAM : SH-4 Tmrd =< 4

The following parameters do not have PC100 symbols, so as reference the Elpida SDRAM notation is used

### 5.2.6.15 $I_{EP}$

$I_{EP}$ is the last data out to precharge minimum delay. This defines **the minimum delay between the last read data being output by the SDRAM and the next PRE command.** Note that this is generally a **negative** parameter which means the precharge can be issued before the last data has been clocked out. The SH-4 BSC will always issue the precharge after the last data has been sampled so this parameter will be met.

### 5.2.6.16 $I_{APR}$

$I_{APR}$ defines **the minimum delay between the last read data being output by the SDRAM and the next ACTV command.** The next ACTV command following sampled data will happen following a precharge in a RAS Down mode read, or following the RAS precharge period (MCR.TPC) in the case of auto precharge read. The worst case when RAS Down is disabled (MCR.RASD=0), and hence the following equation needs to be satisfied:-

SDRAM : SH-4 $I_{APR} =< (MCR.TPC)$

#### 5.2.6.17 $I_{CDD}$

$I_{CDD}$ defines **the delay between CS being negated and SDRAM command reception being disabled.** This will always be 0, as when CS is high, this constitutes a DESL, or deselect. Hence this parameter will always be met.

#### 5.2.6.18 $I_{PEC}$, $I_{BSR}$, $I_{BSH}$ & $I_{BSW}$

These parameters define the timing of PWRDN (power down) and BST (burst stop) commands. Since none of these commands are issued by the SH-4 BSC, none of these parameters need to be met.

### 5.2.7 Auto refresh interval timing

When the SDRAM is in normal operation, auto refreshes are issued by the BSC to keep the memory cells refreshed. Each refresh command will refresh one row at a time. The refresh works by reading the data into the sense amps and then, without outputting them, moving them back into the memory cells. An internal counter inside the SDRAM controls the actual row being refreshed. The particular row number being refreshed is not of interest, as long as every row is refreshed within the refresh period, $t_{REF}$. *When RAS Down mode is used, there is an additional restraint on the refresh period as described below.*

#### 5.2.7.1 $t_{REF}$

$t_{REF}$ is **the longest period any row can go without refreshing, to guarantee data integrity.** Because each refresh cycle will only refresh one row, the Refresh Time Constant Register (RTCOR) and the Refresh Timer Control/Status Register (RTSCR) must be configured to give a refresh period Tras as follows:-

SDRAM : SH-4 $t_{REF}$ => ($t_{AREF}$ * 2^no_rows) **see below if RAS Down mode is used*

#### 5.2.7.2 Tras (2)

When the high speed RAS Down mode is enabled, instead of the SDRAM state machine normally being in the idle state, it remains in the row active state where possible. This significantly speeds up accesses to the same row, as an ACTV command is not necessary. The side effect however is that it is possible that a ROW can be active for a very long time. As Tras defines **the maximum time a row can be active without a precharge**, provision must be made that a command must be issued to the SDRAM within this time. The easiest and safest way of doing this is to set the refresh period to less than Tras. Hence when RAS Down mode is used, the following equation must also be satisfied:-

SDRAM : SH-4 Tras => $t_{AREF}$

### 5.2.8 Clock Quality Parameters

The SDRAMs are normally clocked by the SH-4 CKIO output pin. There are several parameters defined in the SDRAM datasheet, which need to be met by the this clock. The specification of the clock requirements for SDRAMs is very tough to meet. This comes from the PC market place where usually dedicated devices are used to provide extremely high quality clock signals. The SH-4 has a built in PLL (phase locked loop) which is used to match the phase of the buffered CKIO with the internal clock which in some cases is able to drive the CLK input on the SDRAM directly. It is important to study the clock loading carefully however, as applying maximum load to the CKIO pin will mean that some of the SDRAM clock parameters will be difficult to meet, particularly when the CKIO frequency is very high. In these cases clock buffering may be necessary, and to reduce the skew to a minimum, a PLL buffer must be used. Several reference designs are available from Hitachi which give example usage of PLL buffering in large systems.

To aid the understanding of the clock parameters the diagram below has been included. Because different voltage threshold levels are used for different tests, in many cases more margin is often available when performing the timing calculations.

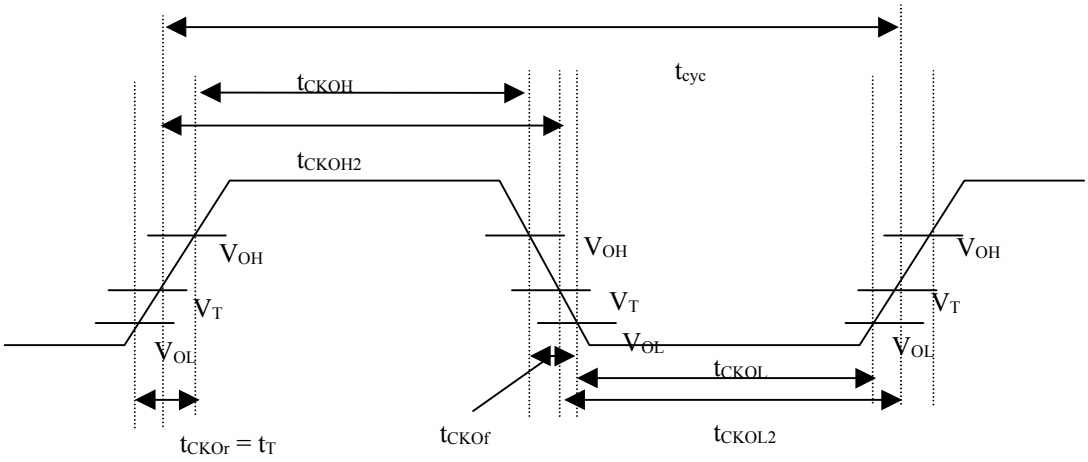


**Figure 5-3: CKIO output and CKE input timing**

#### 5.2.8.1 tT

$t_T$ is the **required input clock rise / fall time**. Actually, in the case of SDRAM, the most important parameter is the clock rise time, as the state machine and hence sampling is synchronous to this edge. Most SDRAMs are tested using $t_T$ = 1ns, so most SDRAM parameters assume the condition $t_T$ = 1ns also. In the case where the rise time is longer than 1ns, sometimes deration factors, sometimes called transient time compensation are added. The factors added (if at all) depend on the manufacturer. This makes it more difficult to meet some of the SDRAM parameters, particularly the ones classified under the “Setup and hold time parameters”.

For example in some cases (tr/2-0.5)ns or [(tr + tf)/2-1]ns should be added to the parameter. For example if the rise/fall time is 4ns, the above equations give 3ns and 1.5ns additional time to some parameters. In this case many parameters can not be met.

The test condition for the SH-4 CKIO output buffer is 30 pF load, which gives a guaranteed rise and fall time of 4ns. Because the slew rate and hence rise / fall time is almost completely dependent on the target, it is important to design the clock distribution very carefully. Track length, characteristic impedance and track topolgy are all important, however the single most important factor for rise and fall time is load capacitance.

Halving the load capacitance can halve the rise time, so this should be taken into consideration during design. In the case of the SH-4/7750, a second CKIO output is available which can make design easier.

Please note that the PLL does not phase compensated CKIO2. It is important to match the loads on CKIO & CKIO2 for minimum skew. The 7750 hardware manual details the internal connections of the CKIO / CKIO2 pins in the appendix.

As clock distribution design is a very specialised skill, it is highly recommended that the designer characterises his own CKIO rise/fall time for the actual target being used. The quality of CKIO rise/fall also influence the setup and hold times and it is recommend to read in this context chapter “Hold Time consideration using slew rate”. Some further details on clock design are provided in the ‘Clock connection guidelines’ section of this application note

**PLL Circuit 2:** PLL circuit 2 coordinates the phases of the bus clock and the CKIO pin output clock. Starting and stopping is controlled by a frequency control register setting. The PLL2 characteristic is not specified in the manual. Hence it follows that a selection of a right external PLL is difficult. Don’t use PLL circuit 2 when using a external PLL.

**DLL Circuit:** This feature is available for SH7751 with CKIO of 66 MHz. As with the PLL circuit 2, the DLL circuit coordinates the phases of the bus clock and CKIO pin output clock. Starting and stopping are controlled by the frequency control register 2 setting. The main difference with the PLL circuit 2 is the stabilisation time for the clock output from the CKIO pin at startup of 6 CKIO cycle. PLL circuit 2 is automatically switched off when the DLL bit is set in the register. Figure 5-4 shows a block diagram of it.

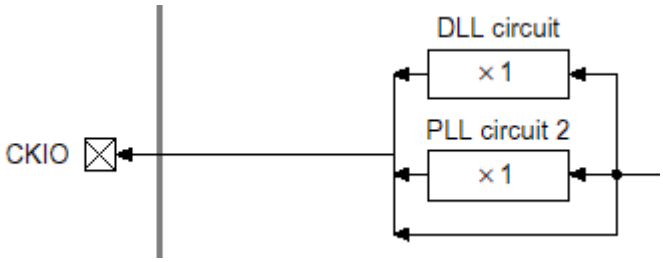


**Figure 5-4: DLL and PLL2 circuit, SH7751**

#### 5.2.8.2 Tclk

Tclk is the **minimum cycle time** for the SDRAM input clock. Hence it also specifies the **maximum frequency** the SDRAM can operate at. Since virtually all SDRAM recommended for design in is at least PC100 specification, even with the fastest CKIO clock, this parameter can be met.

#### 5.2.8.3 Tch/Tcl

Tch are the **minimum clock high/low times** required by the SDRAM. Note that the voltage threshold for this measurement is 1.4V. The SH-4 BSC specification provides two values based on both the normal LVTTL thresholds ($t_{CKOH}/t_{CKOL}$) and 1.5V ($t_{CKOH2}/t_{CKOL2}$). The 1.5V threshold timing should be used for this equation as follows:-

SDRAM : SH-4 $t_{CKH} =< t_{CKOH2}$

SDRAM : SH-4 $t_{CKL} =< t_{CKOL2}$

### 5.2.9 Timing Sheet, SH7750S and 256-Mbit SDRAM

The following timing sheet, table xx, enables you to do all the timing related calculation in a quick and easy way. The applied timings rules are already discussed in the past sections. You could overtake this example into your own exel sheet.

The exel sheet is devided in four main parts, the 'SDRAM', the 'SH7750S', the 'Equation' and the 'SH-4 Bus State Controller Settings'.

The SDRAM timings, e.g. Tckl or Tac, should be referred in the latest data sheet and entered into the columns Min [ns] or Max [ns]. The same have to be done with the SH7750S timings like tCYC or tAD.

The register settings in part 'SH-4 Bus State Controller Settings' are the result of the discussion in the past section. Overtake these settings and adapt it to your requirements. The column '(possible values) Unit' indicates you the allowed settings.

The part 'Equations' lists all the formulas necessary to check and guarantee the SDRAM timings on the left hand side with the timings of the SH-4 on the right. The column 'Meet Timing Constraint?' compares then the SH-4 resulting timings of the column 'Result' with the minimal or maximal SDRAM timings. The entry in column 'Meet Timing Constraint?' is 'TRUE'if the SH-4 meets the SDRAM timing.

| SDRAM:EDS2516AP | PC/100 Symbol | Min [ns] | Max [ns] | CKIO Cycles | Equation | Result [ns] | Meet Timing Constraint?* | |
|---|---|---|---|---|---|---|---|---|
| clock cycle time | **Tclk** | 10 | | 1,0 | =< tCYC | 10,0 | TRUE | |
| access time from rising edge of clock | **Tac** | | 5,4 | 0,5 | =< tCYC-tRDS | 7,0 | TRUE | |
| data output hold past next rising edge | **Toh** | 2,7 | | 0,3 | >= tRDH | 1,5 | TRUE | |
| CLK to Data-out low impedance | **Tlz** | 1 | | 0,1 | =< (A3W-1)*tCYC | 10,0 | TRUE | |
| CLK to Data-out high impedance | **Thz** | | 5,4 | 0,5 | =< A3IW*Tcyc+R22 | 10,0 | TRUE | |
| input setup time for signals: address, data, CS and CE | **Tsi** | 1,5 | | 0,2 | =< tCYC-MAX(setup times) | 4,0 | TRUE | |
| input hold time for ALL signals | **Thi** | 0,8 | | 0,1 | =< MIN(hold times) | 1,5 | TRUE | |
| Refresh or Active to Refresh or Active command period (same bank | **Trc** | 60 | | 6,0 | =< TRAS*Tcyc | 70,0 | TRUE | |
| Active to Precharge command period | **Tras** | 45 | 120000,0 | 4,5 | =< MIN((RCD+A3W+BL),(RCD+BL+TRWL))*tCYC | 80,0 | TRUE | |
| | | | | | =< Taref (see sheet SDRAM Refresh) | 15360,0 | TRUE | |
| Active command to column command (same bank) | **Trcd** | 15 | | 1,5 | =< RCD*tCYC | 20,0 | TRUE | |
| Precharge to active command period | **Trp** | 15 | | 1,5 | =< TPC*tCYC | 20,0 | TRUE | |
| write recovery or data in to precharge lead time | **Tdpl** | 15 | | 1,5 | =< TRWL*tCYC | 20,0 | TRUE | |
| Active to Active command period (different bank) | **Trrd** | 15 | | 1,5 | see section 'Timing Check' | | Always YES | |
| Column command to column command | **Tccd** | 10 | | 1,0 | see section 'Timing Check' | | Always YES | |
| Refresh period | **Tref** | | 64000000,0 | 6400000,0 | | | | |
| self refresh exit time | **Tsrx** | 10 | | 1,0 | =< TRC*tCYC (see section 'Timing Check') | 30,0 | TRUE | |
| last data in to active command (autoprecharge, same bank) | **Iapw/Tdal** | 40 | | 4,0 | =< (TRWL+TPC)*tCYC | 40,0 | TRUE | |
| last data out to active command (autoprecharge, same bank) | **Iapr** | 10 | | 1,0 | =< TPC*tCYC | 20,0 | | |
| Precharge command to high impedance | **Troh** | 20 | | 2,0 | see section 'Timing Check' | | Always YES | |
| Write command to data in latency | **Tdwd** | 0 | | 0,0 | see section 'Timing Check' | | Always YES | |
| DQM to data in | **Tdqm** | 0 | | 0,0 | see section 'Timing Check' | | Always YES | |
| DQM to data out | **Tdqz** | 20 | | 2,0 | see section 'Timing Check' | | Always YES | |
| Registers to active command | **Tmrd** | 20 | | 2,0 | =< 4*tCYC | 40,0 | TRUE | |

| SH7750S | SH4 Symbol | Min [ns] | Max [ns] | | SH4 Bus State Controller Settings | Register Symbol | Value | (possible values) Unit |
|---|---|---|---|---|---|---|---|---|
| CKIO | **tCYC** | 10 | 1000,0 | | **WCR 1** | | | |
| Address delay time | **tAD** | 1,5 | 6,0 | | Idle cycles after SDRAM access | **A3IW** | 1 | 0, 1, 2, 3, 6, 9, 12 or 15 CKIO cycles |
| /BS delay time | **tBSD** | 1,5 | 6,0 | | | | | |
| /CS delay time | **tCSD** | 1,5 | 6,0 | | **WCR2** | | | |
| /RW delay time | **tRWD** | 1,5 | 6,0 | | CAS latency | **A3W** | 2 | 1, 2, 3, 4 or 5 CKIO cycles |
| /RD delay time | **tRSD** | 1,5 | 6,0 | | | | | |
| /WE delay time (falling edge) | **tWEDF** | | 6,0 | | **MCR** | | | |
| /WE delay | **tWED1** | 1,5 | 6,0 | | bank active-read/write command delay time | **RCD** | 2 | 2, 3 or 4 CKIO cycles |
| Write data delay time | **tWDD** | 1,5 | 6,0 | | precharge after refresh | **TRC** | 3 | 0, 3, 6, 9, 12, 15, 18, 21 CKIO cycles |
| /RAS delay time | **tRASD** | 1,5 | 6,0 | | precharge to bank active delay [MCR] | **TPC** | 2 | 1, 2, 3, 4, 5, 6, 7, 8 CKIO cycles |
| /CAS delay time 1 | **tCASD1** | 1,5 | 6,0 | | write precharge delay | **TRWL** | 2 | 1, 2, 3, 4, 5 CKIO cycles |
| /CAS delay time 2 | **tCASD2** | 1,5 | 6,0 | | delay after refresh | **TRAS** | 7 | (4, 5, 6, 7, 8, 9, 10, 11) CKIO cycles |
| CKE delay time | **tCKED** | 1,5 | 6,0 | | Bus width of SDRAM | **SZ** | 64 | bits |
| DQM delay time | **tDQMD** | 1,5 | 6,0 | | | | | |
| /FRAME delay time | **tFMD** | 1,5 | 6,0 | | **Miscellaneous** | | | |
| /ICIOWR delay time (falling edge) | **tICWSDF** | 1,5 | 6,0 | | Burst length of SDRAM | **BL** | 4 | 4 or 8 CKIO cycles |
| /ICIOWR delay time (falling edge) | **tICRSD** | 1,5 | 6,0 | | Bus frequency | **CKIO** | 100 | Mhz |
| DACK delay time | **tDACD** | 1,5 | 6,0 | | Bus cycle time | **tCYC** | 10,0 | ns |
| DACK delay time (falling edge) | **tDACDF** | 1,5 | 6,0 | | | | | |
| Read data hold time | **tRDH** | 1,5 | | | | | | |
| /RDY hold time | **tRDYH** | 1,5 | | | | | | Setup and Hold parameters |
| /IOIS16 hold time | **tIO16H** | 1,5 | | | | | | Wait State Parameter |
| Read data setup time | **tRDS** | 3 | | | | | | do not set in RAS Down Mode |
| /RDY setup time | **tRDYS** | 3 | | | | | | input from data sheet SH4/SDRAM |
| /IOIS16 setup time | **tIO16S** | 3 | | | | | | |

**Table 5-3 Timing Sheet**

## 5.3 Other loading issues

### 5.3.1 General bus loading – output buffer timing

The standard test condition for all SH-4 AC characteristics is $C_L$=30 pF. In fact it is possible to load most pins (not including CKIO) up to 80 pF. The penalty you pay is slower rise times, and hence delayed output signals. This has an impact on the output signal timing. The diagram below shows the timing deration vs. output load for the SH-4. This factor must be applied to all pin timings where 30 pF load is exceeded. In general, the recommendation is to have only SDRAM and buffers directly connected to the SH-4 busses, to avoid more than 30 pF load. This will ensure that bus timings are as easy as possible to meet. The table below shows example load capacitance values for the SH-4, SDRAM, FLASH, buffers and a LCD controller. You can see that a typical system with 2pcs FLASH, 2pcs SDRAM and an LCD controller could easily impose a load of over 40 pF on for example, the address bus.

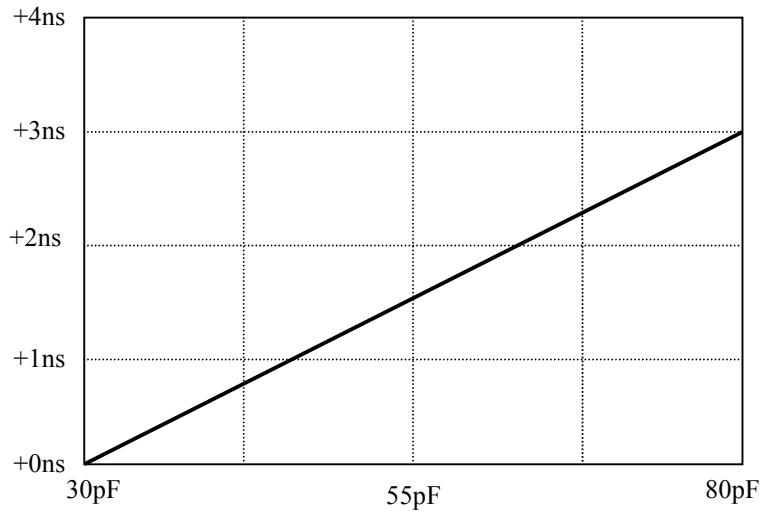


**Figure 5-5: Timing deration versus load capacitance**

| Device | Input Capacitance | Load Capacitance test condition | Bandwidth requirement |
|---|---|---|---|
| SH4/7750 | 10pF Max | 30-80pF | - |
| SDRAM (1pc) | 5pF Max | 50pF | very high |
| FLASH (1pc) | 8pF Max | 30pF | medium |
| LCD controller | 10pF Max | 60pF | medium |
| HD74BC245A | 3pF Typ | 50pF | - |
| PCB track | 3pF (10cm) | - | - |

**Table 5-4: Example load/drive capacitances of different devices**

### 5.3.2 General bus loading – input signal timing

In the case of the SH-4, some of the setup times specified in the AC characteristics are 1.5ns longer for the QFP package than the BGA package. The reason for this is extra capacitance of the leadframe of the QFP package. Care should be taken that the correct figures are used from the data book.

As a general note, most sampled signals, such as RDY and BREQ, are single latched. What this means is that if the setup and hold times are violated, the behaviour of the bus state controller can be a predictable. Therefore the designer must develop hardware that provides a CKIO synchronous input to these pins. This will ensure that setup and hold times are always met, and hence metastability will be avoided.

### 5.3.3 Clock connection guidelines

The load on the CKIO pin should be as low as possible to improve the CKIO rise/fall times. As mentioned in the ‘timing analysis/clock quality’ part of this application not, the CKIO signal routing is perhaps the most critical of any of the SH-4 connections with regard to timing.

Since the CKIO frequency is very high, transmission line effects must be considered and designed for. A suitable textbook (see reference section of this application note) should be consulted for the details. There are also several good modelling tools available which can help optimise the design of transmission lines. This can help you produce an optimal CKIO layout first time round.

The basic idea of the CKIO connection is that the wavefront of the clock edge should be transmitted to each of the receive points, with minimum distortion and skew.

To help ensure this, following points should be followed as closely as possible:

#### 5.3.3.1 Clock topology

Since the wavefront of a signal will travel along a transmission line (here the PCB track) with a finite velocity, the time delay will depend on the length of the track. Because there will usually be at least two receive points and hence the clock track always has a ‘T point’, where the clock track splits and connects each of the receive points. Always try to keep the direction changes to

less than 45 degrees. It is also recommended that length of each of the branches is as small as possible. The distance from the send point to each of the receive points should be equal. Often, this is not possible with a direct track, so sometimes a 'Zig-Zag' is needed to increase the track length to match the other branches. Note that this is also very important when you consider any reflected waves, as these must also meet each other at the T-point on the way back to the SH-4. The diagram below shows an example of this topology.

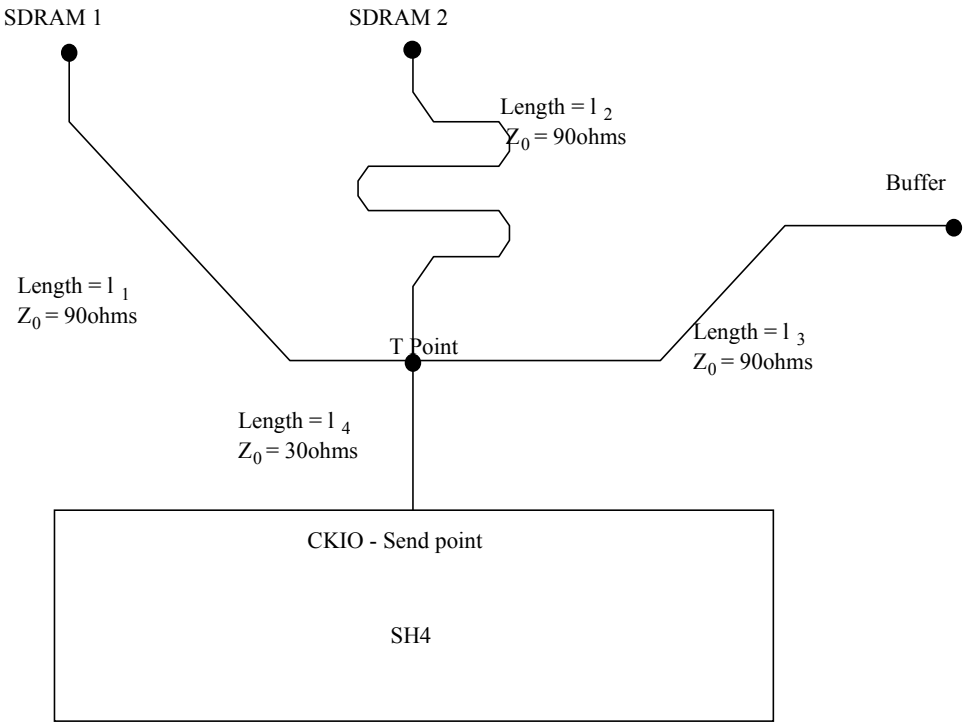


**Figure 5-6: Example CKIO layout**

### 5.3.3.2 CKIO load

The CKIO pin is tested with a load of 30 pF. The guaranteed maximum rise and fall time over the complete operating specification for this condition is 4ns. The actual figure will be better, and with a smaller CKIO load capacitance and good layout, 1-2ns is achievable. It is therefore recommended that the CKIO loading should be very carefully considered at the beginning of the design, and if necessary buffering should be included. In general, 3 loads with good PCB tracking can give a rise/fall time of around 2ns, however to achieve fast rise and fall times with large CKIO loads, it will be necessary to include CKIO buffering. When the skew must be kept to a minimum, it is important to use a PLL (phase locked loop) buffer. An example connection is show below.

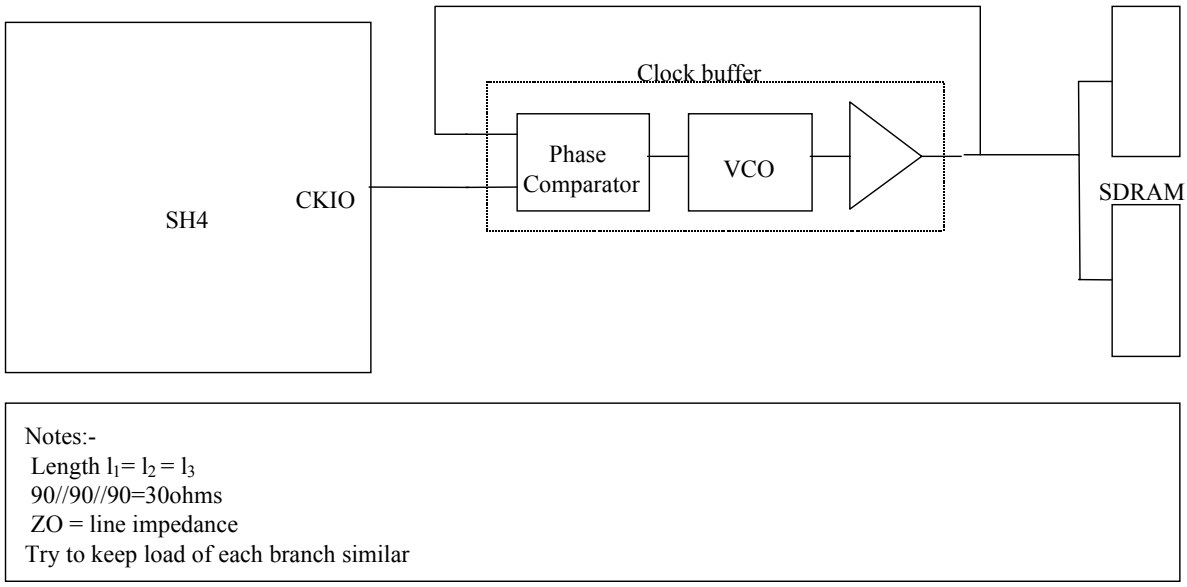


**Figure 5-7: Buffering of CKIO using PLL**

### 5.3.3.3 Characteristic impedance

The approximate output impedance of the CKIO pin is close to 30ohms during the rising edge (approx. 10 ohms for falling edge). The rising edge is most important because it is this that the SDRAM state machine and SH-4/PLL are synchronous to. To optimise the propagation of the clock edge wavefront travelling out towards the clock loads, the characteristic impedance of the clock line should be as close to 30ohms as possible also. Any break in the characteristic impedance will cause reflections. For this reason, vias should be avoided where possible.

Because there will be a T-point where the clock track splits, the wavefront at this point is confronted with parallel loads. To maintain as closely as possible the 30ohm transmission line impedance, the impedance of the clock branches when paralleled together, should equal 30 ohms. Ie. Each branch should have a characteristic of 90ohms if possible where 3 loads exist. This is not a rule that must be followed precisely, however better signal integrity will be achieved if this implemented. Your PCB manufacturer will be able advise on the capabilities of your PCB process.

### 5.3.3.4 Termination

Various types of termination are available for transmission lines. In the case of the SH-4 driving the CKIO line, **it is recommended that no termination should be used**. The reason for this is that series termination at the send point is not needed, as the internal impedance of the SH-4 buffer provides this functionality already. Further impedance added to the clock line would degrade the wavefront of the clock edge too much. Receive point termination should also not be used, as the SDRAMs include built in guaranteed voltage clamps. Further parallel termination (resistive or R-C filter) would impose too much additional load on the CKIO pin (especially in the case of pure resistive), and would also impede the voltage level of the wavefront as it reaches the receive point. Clamp termination at the SDRAM end should not be necessary as SDRAMs

have built in clamp diodes, however it is recommend that any other loads such as buffers, are checked for tolerance to over voltage, and where necessary clamp protected.

As the reflections from each of the receive points will be combined at the T-point, a single wavefront is transmitted back to the CKIO pin. This wavefront will then absorbed by the impedance of the CKIO output buffer, preventing further reflections.

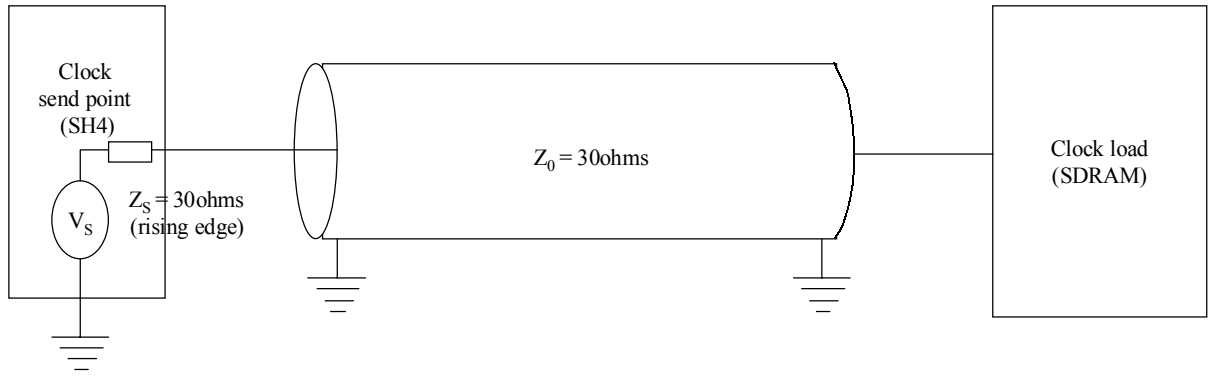


**Figure 5-8: Clock wiring with no termination required**

### 5.3.3.5 Track length

The overall track length of the CKIO net should always be kept as short as possible. The delay times are important, but in addition PCB tracks also have a capacitance value which is generally proportional to length. This can sometimes be as much as 1 pF per 30mm. This should be considered when calculating the total load on the CKIO pin.

It is also very important that any reflected signals arrive back at the CKIO pin before the end of the rise/fall time of the clock. This is so that the internal PLL feedback signal is not falsely clocked. The result is, the total distance from the CKIO send point to each of the receive points (these should all be equal distance away), must be less than 50mm, and as low as 25mm if possible. This is anyhow a good recommendation, as is it will keep the delays (clock skew) down to a few hundreds of picoseconds.

# 6 Worked Example – EDS2516ACTA-7A connected to 2x256-Mbit (64 Mbytes) SDRAM @ 100 MHz bus frequency

In this example 2 x 16-bit wide SDRAMs are connected to the SH7750S, with a bus clock frequency of 100 MHz. The SDRAMs used are the Elpida EDS2516ACTA-7A. It is a 133 MHz device with a CAS latency 2/3 and has a capacity of 256-Mbit SDRAM organised as 4194304 x 16-bit x 4 bank.

This worked example is split into 3 sections: Physical connection, Timing Check and software initialisation.

Only the SDRAM (area 3) will be initialised here, and all other areas settings have been left as default.

## 6.1 Physical connection

The system bus in this example will be partitioned such that the only devices connected directly to the SH-4 pins are the 2 SDRAMs and a buffer. This will reduce the load on most pins to:

2 x 5 pF + 3 pF + track capacitance = Approx. 15 pF total, with 7cm of tracking

This is particularly important for the CKIO connection and will ensure good rise and fall times when combined with a good clock layout. In this case, all of the signals from the SDRAM will connect directly to the SH-4. The most important item to check when checking the physical connection is the address multiplexing. The table below shows how the address lines will be connected during the RAS (ACTV command) and CAS (read/write command) phases of an access.

Note that because the total memory is organised as a 32-bit word, the SDRAM address connections are shifted by 2 bits (8 bits x $2^2$ = 32 bits), which means A0 and A1 are left unconnected on the SH-4. If a 64-bit interface were selected then the address connections are shifted by a further one bit. Table 6-1 below shows the required address multiplexing.

| SH4 address | 13 bit | 9 bit | SDRAM | Function |
|---|---|---|---|---|
| | RAS cycle | CAS cycle | Address Pins | |
| A16 | A25 | A25 | A14 | BANK select address 1/0 |
| A15 | A24 | A24 | A13 | |
| A14 | A23 | X | A12 | |
| A13 | A22 | X | A11 | |
| A12 | A21 | H/L | A10 | Address / precharge setting |
| A11 | A20 | X | A9 | Address |
| A10 | A19 | A10 | A8 | |
| A9 | A18 | A9 | A7 | |
| A8 | A17 | A8 | A6 | |
| A7 | A16 | A7 | A5 | |
| A6 | A15 | A6 | A4 | |
| A5 | A14 | A5 | A3 | |
| A4 | A13 | A4 | A2 | |
| A3 | A12 | A3 | A1 | |
| A2 | A11 | A2 | A0 | |
| A1 | X | X | Not used | |
| A0 | X | X | Not used | |

**Table 6-1: Address multiplex table**

The table was derived as follows:

The device has a **9 bit column address** (which are the lowest 9 bits of the address, corresponds to 512 columns) and these must connect properly during the CAS cycle. The box shows the first 9 bits of the address correctly connecting during the CAS cycle, with SH-4/A2 – A10 connecting to SDRAM/A0 – A8.

The **13 bit row address** (the next 13 higher bits of the address, corresponds to 8192 rows) will be output during the RAS cycle. SH-4/A11 – A23 must be output to SDRAM/A0 – A12, as shown by the box on the RAS column.

Next the two bank bits (the highest two address bits, corresponds to 4 banks) must connect correctly to the SH-4/A24 + A25, and this must happen during both RAS and CAS cycles.

Finally, the address precharge bit must be correctly output during the CAS cycle. On this SDRAM, A10 is the address precharge bit so this must be correctly output from the SH-4/A12. This bit makes up part of the command word and differentiates between a precharge or RAS Down access.

The table above matches the SH-4/7750S multiplex setting AMX 6, AMXEXT 1. This gives us the AMX settings in the MCR:

Multiplex setting AMX 6, AMXEXT 1:

MCR.AMX = $110_2$ ( for the representation of the bit settings in the BSC the assumption is made that the register are declared in C-structures)

MCR.AMEXT = $1_2$

The resulting connection between the SH-4 and the SDRAMs will be as follows:

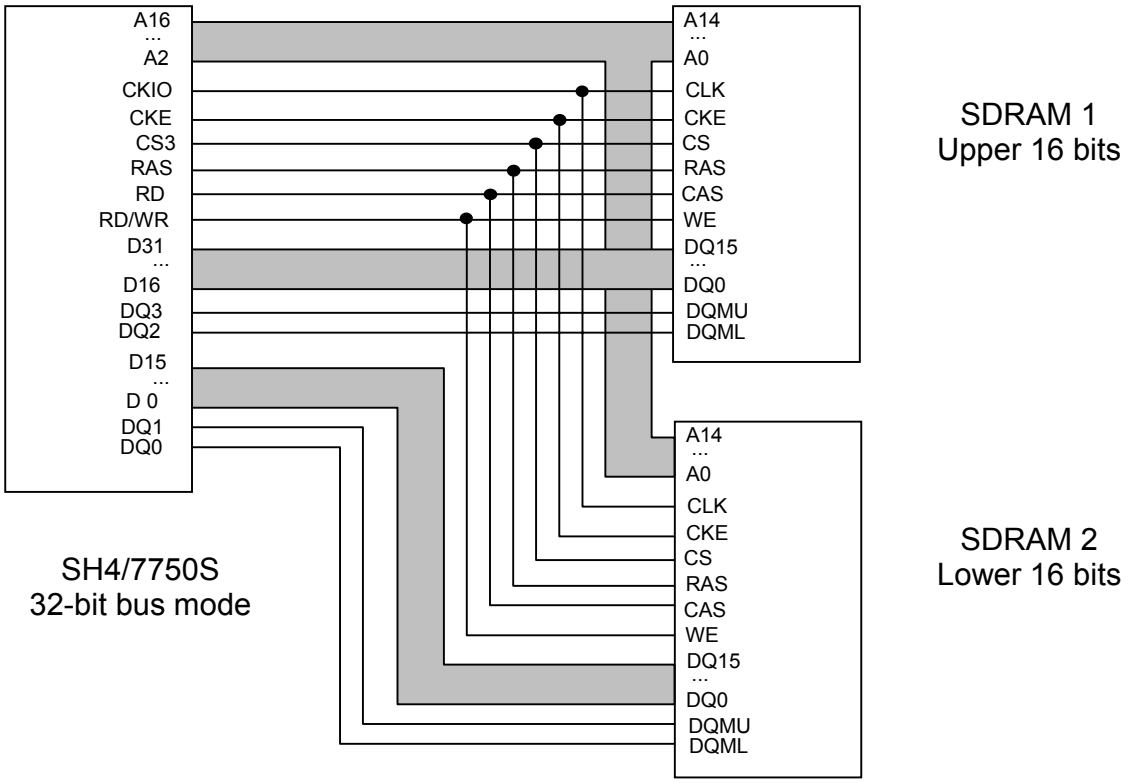


**Figure 6-1: Physical connection**

## 6.2 Timing Check

Through this section, each of the timing parameters described earlier in this application note will be satisfied and the appropriate SDRAM control registers values will be derived.

The already discussed timing sheet in section '5.2.9 Timing Sheet, SH7750S and 256-Mbit SDRAM' contains already all timing information, equation and register settings for this working example and should be used as quick interface timing guide for different SDRAMs.

Only the parameters which specify an equation to meet will be considered, as all other parameters have already been met. The first decision to be made has already happened – the bus frequency will be **100 MHz**, and hence the cycle time will be 10ns. For the purpose of brevity, **a cycle time of 10ns** will be used, which also provides a small amount of extra margin for timing calculations.

The EDS2516ACTA-7A grade is capable of running at 133 MHz with CAS latency 2 or 3. A CAS latency of 2 is chosen which improves in general the overall performance.

This gives us our next initialisation value.

CAS latency 2: WCR2.A3W = $010_2$

Note: Each bus frequency has a different set of timing parameters. Make sure that the correct figures from the databook are used during the timing check. In this case the 100 MHz bus figures have been used.

### 6.2.1 Setup and Hold parameters

| Item | SDRAM | | | SH4 | | | Ok | Notes |
|---|---|---|---|---|---|---|---|---|
| | | min | max | | min | max | | |
| Access time from CLK | Tac | - | 5,4 | >= tCYC- <= tRDH | - | 7 | Yes | |
| Data output hold time | Toh | 2,7 | - | | 1,5 | - | Yes | |
| Input setup time | Tsi | 1,5 | - | >= tCYC-txxS | 4 | - | Yes | |
| Input hold time | Thi | 0,8 | - | >= txxD | 1,5 | - | Yes | This value is typically 1.5ns for SH4, however 1ns is safe to use |
| CLK - data low impedance | $t_{LZ}$ | 1 | - | >= (A3W-1)*tCYC | 10 | - | Yes | CAS latency 2. The idle settings of other CS areas must be adjusted to meet this |
| CLK - data high impedanc | $t_{HZ}$ | - | 5,4 | >= A3IW*tCYC | - | 10 | Yes | An idle setting of 0 might be possible in WCR1 for area CS3 |

**Table 6-2: Setup and Hold parameters**

The SDRAM side of the table shows what value the SDRAM needs, and the SH-4 section shows the value that the SH-4 can provide from the equation, using values from the databook. In all cases the values can be met, which means that all data in/out operations and command issuance

to the SDRAM can be performed correctly. There are 2 notes which have been included which are as follows:

The CLK to low impedance time will depend on CAS latency settings and hold time of other devices in the system. We will assume a worst case CAS latency of 2, which assuming a system worst case system hold of 2ns gives 12ns margin.

Following a SDRAM access, the SDRAM will hold the data bus for maximum 5.4ns. If other devices in the system can assert low impedance on the bus within 5.4ns, an idle cycle will be required in WCR1 for area 3. We will assume that other devices in the system can assert the databus within 5.4ns of being selected, so the idle cycle setting is 1 for area 3. This gives us the initialisation value for WCR2:

0 Idle cycle insertion: WCR1.A3IW = $001_2$

If there isn't a device which can assert the bus faster then 5.4ns also the idle cycle setting of 0 for area 3 might be possible.

### 6.2.2 Wait State Parameters

The wait state parameters can be controlled by adjusting the BSC registers. The goal is to minimise all register values to minimum wait times, whilst still meeting every timing parameter. *Note that in some cases, multiple parameters depend on a single register setting.* In this case the smallest value which meets all of parameters should be chosen. Only the parameters that an equation has been specified for are considered, as the rest have already been met.

| | SDRAM | Unit | SH4 | | Unit | Ok | Note |
|---|---|---|---|---|---|---|---|
| Trcd | 15 | ns | >=tCYC*MCR.RCD | 20 | ns | Yes | MCR.RCD = 2 |
| Trp | 15 | ns | >=tCYC*MCR.TPC | 20 | ns | Yes | MCR.TPC = 2 |
| Tdpl | 15 | ns | >=tCYC*MCR.TRWL | 20 | ns | Yes | MCR.TRWL = 2 |
| Trc | 60 | ns | >=tCYC*MCR.TRAS | 70 | ns | Yes | MCR.TRAS = 0 & MCR.TRC = 3 |
| Tdal | 4 | cyc | (MCR.TRWL + MCR.TPC) | 4 | cyc | Yes | TPC,TRWL = 2 |
| ISREX | 1 | cyc | MCR.TRC | 3 | cyc | Yes | MCR.TRC = 3 |
| Tmrd | 2 | cyc | 4 (fixed on SH4) | 4 | cyc | Yes | |
| IAPR | 1 | cyc | MCR.TPC | 2 | cyc | Yes | MCR.TPC = 2 |

**Table 6-3: Wait State Parameters**

Deriving the various MCR settings is an iterative process. It is best to start with the smallest setting first, and work upwards. TRC is increments in values of 3, so 3 is the minimum setting. As this delay is only inserted during a refresh cycle, not providing an optimal setting here is not significant. TRAS is a constant (4 to 11) + TRC, and as TRC is already 3, the minimum setting can be made and Trcd is easily met. TRWL has to be increased to 2 to satisfy Tdpl so the following settings in the MCR have been derived:

| MCR Register bits | Binary value | CKIO cycles |
|---|---|---|
| MCR.TPC | 001 | 2 |
| MCR.RCD | 01 | 2 |
| MCR.TRWL | 001 | 2 |
| MCR.TRC | 001 | 3 |
| MCR.TRAS | 000 | 0 |

**Table 6-4: MCR settings**

So far we have defined most of MCR, and the settings are summarised in the diagram below:

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RASD | MRSET | TRC2 | TRC1 | TRC0 | - | - | - | TCAS | - | TPC2 | TPC1 | TPC0 | - | RCD1 | RCD0 |
| Setting | 1 | 0* | 0 | 0 | 1 | 0 | 0 | 0 | X (0) | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | TRWL2 | TRWL1 | TRWL0 | TRAS2 | TRAS1 | TRAS0 | BE | SZ1 | SZ0 | AMEXT | AMX2 | AMX1 | AMX0 | RFSH | RMODE | EDO |
| Setting | 0 | 0 | 1 | 0 | 0 | 0 | X (0) | 1 | 1 | 1 | 1 | 1 | 0 | 0* | 0* | X (0) |

**Table 6-5: Memory Control Register (MCR)**

The remaining settings not described above are as follows:

RAS Down – This will enable the high speed access mode
*MRSET. This is used during the initialisation. See the code later for details
SZ. Memory width setting. This is 32 bits (= 11) as mentioned in the worked example title.
*RFSH. This enables refreshing and is demonstrated in the attached code.
*RMODE. This is used for putting the SDRAM into self refresh and is not used in this example.

The above basic MCR setting will be used in the code example, along with the bits with an asterisk, to initialise the SDRAM controller.

### 6.2.3 Auto Refresh Interval parameters

There are two constraints on the refresh time here, and the lowest of the two must be chosen. The maximum row active time (Tras) here is 120us, and the self refresh command period is as follows

$t_{AREF}$ = $t_{REF}$ / 2 ^ no_rows = 64ms / 2^13 = 7.8125 µs

As this value is smaller, this is the value we must use for the refresh generator. To calculate the values needed for the refresh registers, the first thing to define is how many CKIO cycles fit into 7.8 µs.

This is 7.8125us / 10ns = 781.25 (= 781 cycles when rounded down).

Because the smallest prescaler value the refresh counter can be driven with is CKIO/4, this corresponds to a refresh time constant register (RTCOR) setting of 781/4 = 195. Hence:

Setting for refresh time constant register: RTCOR = 195d

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | - | - | - | - | - | - | - | - | CMF | CMIE | CKS2 | CKS1 | CKS0 | OVF | OVIE | LMTS |
| Setting | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | X (0) | 0 | 0 | 0 | 1 | X (0) | 0 | X (0) |

**Table 6-6 RTSCR setting**

From here we can now define the RTSCR setting also:The settings are as follows:

bits 15-8 Reserved 0

**C**ompare **M**atch **F**lag 0 Status flag. Don't care.

**C**ompare **M**atch Interrupt **E**nable. Set to 0 to disable it

**C**lock **S**elect **B**its = **0**, **0**, **1** Refresh count clock is CKIO/4

**R**efresh Count O**v**erflow Status **F**lag. Don't care.

**R**efresh Counter O**v**erflow **I**nterrupt **E**nable. Set to 0 to disable it.

Refresh Count Overflow **Li**m**it** **S**elect. Don't care

### 6.2.4 Clock quality parameters

| | SDRAM | unit | SH4 | | unit | Ok | note |
|---|---|---|---|---|---|---|---|
| tT | 0,5 | ns | tT | <1 | ns | Yes | This value is target characterized |
| Tch | 2,5 | ns | tCKOH2 | 3 | ns | Yes | uses midpoint threshold |
| Tcl | 2,5 | ns | tCKOL2 | 3 | ns | Yes | uses midpoint threshold |

**Table 6-7: Clock quality parameters**

As mentioned in the clock connection guidelines section of this application note, clock rise and fall time is highly dependent on the target system. The rise time of an unloaded CKIO pin is much less than 1ns, however the test condition of $C_L$=30 pF gives a guaranteed maximum rise time of 3ns. Following the guidelines in the clock connection guidelines will ensure optimum rise and fall times, however in cases of heavy CKIO loading, external PLL buffering will be required and is which finally guarantees that the tT of the SDRAM is met.

In this example, the CKIO pin is connected to 2 SDRAMs and a standard buffer. This will reduce the load approx. 15 pF total, including tracking, and will help produce a good CKIO risetime.

## 6.3 Producing the initialisation code

The above sections have proved that the electrical timing of the SDRAM can be met, and consequently derived the values that need to be initialised into the BSC. The next step is to produce the initialisation code for the SDRAM, based on these values. The initialisation flow used in this application note is the one recommended in the HW manual. Normally the SDRAM is the only RAM available in the system, so until the SDRAM is initialised, stack cannot be used. Since C compilers have the possibility to use the stack, care must be taken if the code is written in C. Function calls are using the stack and therefore not allowed expect the RAM of the operand cache is available during booting. The alternative and more often seen method is to write your own code in assembler.

For this example, C code is used using the SH/7750S IO definition file that is generated by Hitachi Embedded Workshop (HEW). This file describes the peripheral IO registers as bit fields, and hence produces very readable code. It also means that only the required bits are modified, with others being left to their present value. Hence each modified bit field will generate a read modify write mask operation. Write only registers or password protected cannot be used manipulated in this way, hence the SDMR and refresh registers are initialised using a full register load.The initialisation sequence is as follows:

Initialise BCR1 to define *memory type* (Ie. Area 3 will be SDRAM)

Initialise BCR2 to define *memory widths*

Initialise WCR1,2 & 3 to define *idle cycles, CAS* and also *wait states* for non SDRAM areas

Initialise the *refresh controller* (RTCSR, RFCR, RTCNT & RTCOR). Note that refreshing will not take place yet because it has not been enabled in the MCR

Initialise the MCR with the wait state settings as derived above. Note that bits 30 (MRS), 2 (RFSH) & 2 (RMODE) should be set to 0, to disable mode register set (i.e. enable PALL) and refreshing.

Next, the PALL (precharge all) command should be sent to the SDRAM in area 3. This is done by writing a byte to address H'FF940000 + SDMR value shifted left by 2 bits.

Eight REF (auto refresh) commands should now be sent to the SDRAM. This is done by reducing the refresh interval to a minimum, and waiting until the refresh count register (RFCR) is greater than 7. The refresh control bit (MCR.RFSH) must be set to one prior to this to enable refreshing.

Next the MRS (mode register set) command should be sent to the SDRAM. This initialises the internal configuration register (SDMR) of each of the SDRAMs connected. To do this, bit MCR.MRS is set to 1, followed by a write to a special address according to CAS latency, bus width and chip select area. In this case the CAS latency is 2, the bus width is 32 bits and the chip select area is 3. Hence a byte is written to H'FF94008C. Note that the actual value written does not matter – it is the address which defines the write value to the SDMR

SDRAM is ready for use!

The code to perform this procedure, using the previously calculated values is as follows:

### 6.3.1 Example SDRAM initialisation code

```
#include "iodefine.h"        /*7750S standard HEW IO define file */

#define SDMR_VAL 0x008C  /*Burst length 8 (32byte/32bit), CAS latency 2, sequential wrap type*/

#define SDMR3  (*(volatile char *)(0xFF940000 + SDMR_VAL))

              /*Address calculation for MRS command*/

#define RTPWD1 0xA500        /*Refresh controller access password for RTCSR, RTCNT, or RTCOR  */
#define RTPWD2 0xA400        /*Refresh controller access password for RFCR        */

void sdram_setup(void)
{
BSC.BCR1.BIT.DRAMTP=2;         /*Area 3=SDRAM, Area 2=SRAM    */

BSC.WCR1.BIT.A3IW=1;       /*one idle cycles for area 3    */
BSC.WCR2.BIT.A3W=2;        /*CAS latency for area 3=2 cycles    */
BSC.MCR.BIT.RASD=1;        /*RAS Down mode enabled      */
BSC.MCR.BIT.MRSET=0;       /*Enable output of PALL command    */
BSC.MCR.BIT.TRC=1;         /*3 cycles RAS precharge after refr.  */
BSC.MCR.BIT.TPC=1;         /*2 cycles RAS precharge period    */
BSC.MCR.BIT.RCD=1;         /*2 cycles RAS-CAS delay      */
BSC.MCR.BIT.TRWL=1;        /*2 cycles write precharge delay    */
BSC.MCR.BIT.TRAS=0;        /*4+TRC interval at end of refresh  */
BSC.MCR.BIT.SZ=3;          /*32bit wide SDRAM bus      */
BSC.MCR.BIT.AMXEXT=1;      /*AMX setting for 2x256-Mbit SDRAM...  */
BSC.MCR.BIT.AMX=6;         /*..9bit column, 13bit row + 4 bank  */
BSC.MCR.BIT.RFSH=0;        /*switch refreshing off for now    */
BSC.MCR.BIT.RMODE=0;       /*Select auto refreshing, not self  */
BSC.RTCSR.WORD=(RTPWD1 & 0x0008); /*Overflow + compare match interrupt disabled, clock = bus phi/4
count limit = dont care      */
BSC.RTCNT=(RTPWD1 & 0);         /*Clear timer counter      */
BSC.RTCOR=(RTPWD1 & 1);           /*Set compare match value to 1 decimal This will enable 8 fast
refreshes    */
BSC.RTCNT=(RTPWD2 & 0);         /*Set refresh count to 0      */
            /*Assume 200us idle time for SDRAM has been satisfied by power on reset  */
SDMR3=0;         /*Precharge all banks (PALL)    */
BSC.MCR.BIT.RFSH=1;         /*enable refreshing      */
while (BSC.RTCNT<8);         /*wait for 8 auto refreshes    */
BSC.MCR.BIT.MRSET=0;         /*Enable output of MRSET command    */
SDMR3=0;         /*Trigger MRSET command      */
BSC.RTCOR=(RTPWD1 & 195);       /*Set compare match value to 195 dec. This returns the interval to
normal  */
            /*SDRAM now set up!      */
}
```

# 7 Summary

The application note shows in detail all important aspects of interfacing SDRAMs to SH-4. From understanding how the SDRAMs works; the physical connection and the timing consideration to writing the initialization code for SH-4. This also forms the base for interfacing new products of the SH-4 familiy: the SH775xR will allow higher data bandwidth due to higher frequency.

# 8 References

SH7751 Hardware Manual V2.0
Hitachi Ltd.

Technical Note; Clock wiring between SH-4 and SDRAM
Hitachi Ltd.

SH7750, SH7750S Hardware Manual V5.0
Hitachi Ltd.

High Speed Digital Design - A Handbook of black magic
Howard Johnson
Martin Graham
ISBN 0-13-395724-1

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.
2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi's permission.
3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during the operation of the user's unit according to this document.
4. Circuitry and other examples described herein are meant only to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples therein.
5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.
6. MEDICAL APPLICATIONS: Hitachi's products are not authorized for use in MEDICAL APPLICATIONS without the written consent of the appropriate officer of Hitachi's sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi's products are requested to notify the relevant sales office when planning to use the products in MEDICAL APPLICATIONS.

Copyright ©Hitachi, Ltd., 2002. All rights reserved